

LEVEL

ARPA ORDER NO. 2223

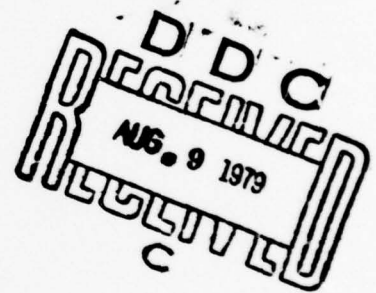
ISI/RR-79-76
May 1979



Victor R. Lesser
University of Massachusetts

Lee D. Erman
USC Information Sciences Institute

An Experiment in Distributed Interpretation



DDC FILE COPY

This document has been approved
for public release and sale; its
distribution is unlimited.

INFORMATION SCIENCES INSTITUTE

UNIVERSITY OF SOUTHERN CALIFORNIA



4676 Admiralty Way/Marina del Rey/California 90291
(213) 822-1511

79 08 8 001

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 14 ISI/RR-79-76	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 An Experiment in Distributed Interpretation	5. TYPE OF REPORT & PERIOD COVERED 9 Research report	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 Victor R. Lesser and Lee D. Erman	8. CONTRACT OR GRANT NUMBER(s) 15 DAHC 15-72-C-0308, VARPA Order-2223	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 12 50P.
9. PERFORMING ORGANIZATION NAME AND ADDRESS ✓ USC/Information Sciences Institute 4676 Admiralty Way Marina del Rey, CA 90291	11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Blvd. Arlington, VA 22209	12. REPORT DATE 14 May 1979
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	13. NUMBER OF PAGES 51	15. SECURITY CLASS. (of this report) Unclassified
16. DISTRIBUTION STATEMENT (of this Report) This document is approved for public release and sale; distribution is unlimited.		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) -----		
18. SUPPLEMENTARY NOTES This report is being published simultaneously by USC/Information Sciences Institute and Carnegie-Mellon University (as CMU-CS-79-120).		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) distributed AI; distributed processing; distributed sensor networks		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) (OVER) 407952 JED		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

79 08 8 001

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. ABSTRACT

↓ The range of application areas to which distributed processing has been applied is limited. In order to extend this range, new models for organizing distributed systems must be developed.

We present a new model, in which the distributed system is able to function effectively even though processing nodes have inconsistent and incomplete views of the data bases necessary for their computations. This model differs from conventional approaches in its emphasis on dealing with distribution-caused uncertainty and errors in control, data, and algorithm as an integral part of the network problem-solving process.

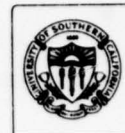
We show how this new model can be applied to the problem of distributed interpretation. Experimental results with an actual interpretation system support these ideas.

←

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DDC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	<input type="checkbox"/>
By _____	
Distribution/	
Availability Codes	
Dist	Availability for special

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)



Victor R. Lesser
University of Massachusetts

Lee D. Erman
USC Information Sciences Institute

An Experiment in Distributed Interpretation

UNIVERSITY OF SOUTHERN CALIFORNIA



INFORMATION SCIENCES INSTITUTE

4676 Admiralty Way/Marina del Rey/California 90291
(213) 822-1511

This research was supported in part by Defense Advanced Research Projects Agency contract F44620-73-C-0074 to Carnegie-Mellon University, National Science Foundation grant MCS78-04212 to University of Massachusetts, and DARPA contract DAHC 1572-C-0308 to University of Southern California Information Sciences Institute. Views and conclusions contained in this document are the authors' and should not be interpreted as representing the official opinion or policy of DARPA, the U.S. Government, or any other person or agency connected with them.

ABSTRACT

The range of application areas to which distributed processing has been applied effectively is limited. In order to extend this range, new models for organizing distributed systems must be developed.

We present a new model, in which the distributed system is able to function effectively even though processing nodes have inconsistent and incomplete views of the data bases necessary for their computations. This model differs from conventional approaches in its emphasis on dealing with distribution-caused uncertainty and errors in control, data, and algorithm as an integral part of the network problem-solving process.

We show how this new model can be applied to the problem of distributed interpretation. Experimental results with an actual interpretation system support these ideas.

This report is being published simultaneously by USC/Information Sciences Institute (as RR-79-76) and Carnegie-Mellon University (as CMU-CS-79-120).

CONTENTS

1. INTRODUCTION	1
2. OVERVIEW OF HEARSAY-II: A System that Handles Uncertainty	3
2.1. The Model	3
2.2. The Architecture	4
3. ISSUES IN DISTRIBUTING HEARSAY-II	7
4. A NETWORK OF HEARSAY-II SYSTEMS	10
4.1. Intranode Considerations -- Selection and Focusing of KSs	11
4.2. Network Configurations	14
4.3. Internode Communication -- Mechanism	16
4.4. Internode Communication -- Policies	16
4.4.1. The Basic Policy	20
4.4.2. Variants	20
5. THE EXPERIMENT	23
5.1. Simulating a Network	23
5.2. Selection of KSs and Areas-of-Interest	23
5.3. Communication Strategy	26
5.4. Knowledge-Source Changes	27
6. RESULTS	29
6.1. Network versus Centralized	30
6.2. Transmission Policies	37
6.3. Communication with Errors	37
7. CONCLUSIONS	39
ACKNOWLEDGMENTS	45
REFERENCES	45

1. INTRODUCTION

An interpretation system accepts a set of signals from some environment and produces higher-level descriptions of objects and events in the environment. Speech and image understanding, medical diagnosis, determination of molecular structure, and geological surveying are problems that have been pursued with interpretation systems. A *distributed* interpretation system may be needed for applications in which sensors for collecting the environmental data are widely distributed, interpretation requires data from at least several of the sensors, and communication of all sensory data to a centralized site is undesirable. Sensor networks (composed of low-power radar, acoustic, or optical detectors; seismometers; hydrophones; etc.), network (automotive) traffic control, inventory control (e.g., car rentals), power network grids, and tasks using mobile robots are examples of potential applications for distributed interpretation. In these applications, an architecture that locates processing capability at the sensor sites and that requires only limited communication among the processors is especially advantageous and is perhaps the only way to meet demands of real-time response, limited communication bandwidth, and reliability.

Two major questions arise in the distributed interpretation task: how to interpret the signal data and how to decompose a given interpretation technique for distribution. Some interpretation algorithms and control structures cannot be replicated or partitioned on the basis of the distribution of the sensory data without requiring unacceptably large amounts of interprocessor communication to maintain completeness and consistency among the local databases. In such a case, it is necessary to modify the algorithm and control structure to operate on local databases that are incomplete and possibly inconsistent. For some interpretation techniques, such modifications might be difficult or impossible.

Knowledge-based Artificial Intelligence (AI) interpretation systems developed recently for speech, image, and signal interpretation applications have structures that seem to make them suitable for decomposition in distributed environments where local databases are incomplete and possibly inconsistent. Examples of these systems include Hearsay-II [Lesser 75, Erman 79], HARP [Lowerre 79], MSYS [Barrow 76], SIAP [Drazovich 78], CRYSLIS [Engelmore 77], and VISIONS [Hanson 78]. These interpretation techniques use the problem-solving paradigm of searching for an overall solution by the *incremental aggregation of partial solutions*. In this paradigm, errors and uncertainty from input data and from incomplete or incorrect knowledge are handled as an *integral* part of the interpretation process. This is in contrast to more conventional problem-solving techniques, in which errors are fatal or are handled as exceptional conditions, requiring additional processing outside the normal problem-solving strategy.

We hypothesize that these knowledge-based AI systems can handle the additional uncertainty introduced by a distributed decomposition without extensive modification.¹ Preliminary work in

¹A more detailed discussion of these points and the appropriateness of knowledge-based AI as the basis for distributed problem-solving systems is contained in [Lesser 78]

testing this hypothesis with respect to synchronization has been encouraging. Experiments with a multiprocessor implementation of the Hearsay-II speech understanding system have shown that eliminating explicit synchronization results in increased parallelism without a decrease in problem-solving accuracy [Fennell 77]. Similarly, a class of iterative refinement methods (although not knowledge-based) for solving partial differential equations has been decomposed for multiprocessor implementation so as to avoid most explicit synchronization, thus allowing for increased speed-up due to parallel processing [Baudet 76]. This decomposition is accomplished by allowing each point in the differential grid to be calculated from values of its neighboring points that are not necessarily the most up-to-date.

While such AI systems provide a promising basis for distributed problem-solving, none has yet been built for a fully distributed environment; centralized global knowledge or global control has been used in existing interpretation systems to coordinate various system modules. In this report, we describe an experiment in the complete decomposition of an existing knowledge-based interpretation model -- Hearsay-II [Erman 75, Lesser 77]. Although Hearsay-II was developed in the context of speech understanding [Lesser 75, Erman 79], its basic structure has been applied to a range of interpretation tasks, including multisensor signal interpretation [Nii 78], protein-crystallographic analysis [Engelmore 77], and image understanding [Hanson 78].

This report concentrates on applying the Hearsay-II architecture to the distributed interpretation problem, where each processor can be mobile, has a set of (possibly non-uniform) sensing devices, and interacts with nearby processors through a packet-radio communication network. Processors communicate among themselves to generate a consistent interpretation of "what is happening" in the environment being sensed.

Section 2 presents a brief overview of the Hearsay-II model of knowledge-based AI interpretation, followed by a description of the Hearsay-II architecture. This section presents mechanisms for handling uncertainty as an integral part of the problem-solving process. This sets the stage for later discussion of how these uncertainty-resolving mechanisms can also be used to resolve uncertainty introduced through distribution of the system. Section 3 outlines several possible directions for designing a distributed Hearsay-II system, with Section 4 presenting the particular design we have adopted here. Sections 5 and 6 describe the distributed Hearsay-II speech understanding system experiment and results; in this experiment, the microphone sensor at each node of the distributed network samples one time-contiguous segment of the speech signal. Finally, some discussion and summary is presented.

Our goal is not to prove that one *should* design a distributed speech understanding system, but rather to point out some of the issues involved in designing a distributed interpretation system dealing with incomplete and inconsistent local data as an integral part of its processing. We are using the Hearsay-II speech understanding system because it has a structure that we feel is appropriate and because it is a large, knowledge-based interpretation system to which we have access. There are serious problems with using this system for experimentation:

- Because of several considerations, discussed in Secs. 5.2 and 6.1, networks are limited to about three nodes.
- Because of the costs of the network simulation, only a limited number of experimental runs could be done and with relatively simple test data and communication policies.
- There is probably no practical need for distributing a single-speaker speech understanding system.

We feel that these limitations are sufficiently outweighed by the advantages of experimentation with a *real* system to make the effort worthwhile and the results, while not conclusive, indicative.

2. OVERVIEW OF HEARSAY-II: A System that Handles Uncertainty

2.1. The Model

We will take as the competence goal of an interpretation system the construction of the most credible complete interpretation of the input data.¹ In Hearsay-II, an interpretation is constructed by combining partial interpretations derived from diverse knowledge. Each area of knowledge is represented by an independent module called a "knowledge source" (KS). In the application of Hearsay-II to speech understanding, for example, these KSs cover such areas of knowledge as acoustics, phonetics, syntax, and semantics. The Hearsay-II architecture is designed to permit cooperative and competitive problem-solving among the KSs in order to resolve the uncertainty caused by noise and incompleteness in the input data and inaccurate processing by the KSs.

The interaction of KSs is based on an iterative, data-directed form of the hypothesize-and-test paradigm. In this paradigm, an iteration involves the creation of an hypothesis, one possible interpretation of some part of the solution, followed by test(s) of its plausibility. When performing these actions, KSs use a priori knowledge about the problem, as well as previously generated hypotheses, which form a context for applying the knowledge. When a KS creates an hypothesis from previously created hypotheses, the KS extends the existing (partial) interpretation with more information, thereby reducing the uncertainty of the interpretation. The processing is terminated when a consistent hypothesis is generated that satisfies the requirements of a complete solution.

A KS often generates incorrect hypotheses because its knowledge or its input data, including previously-generated hypotheses, contains errors or is incomplete. Thus, if KSs were to generate only a single hypothesis for each specific part of the problem, the problem-solving process would often terminate with an inaccurate interpretation or with a partial interpretation that could not be further enlarged because it is inconsistent. In order to avoid this problem, KSs in general create several *alternative* hypotheses for each part of the problem. The KS associates with each hypothesis a *credibility* rating, which is its estimate of the likelihood that the hypothesis is correct. The lower the credibility of the alternatives, the greater the number that must be

¹In general, some applications might not contain a notion of a "complete" or spanning interpretation, but rather be interested in successive partial interpretations. Nothing in the discussion that follows is actually specific to complete interpretations, but we adopt that notion because of our involvement with the speech understanding task and the acceptance of individual, single-sentence utterances.

generated to produce the same likelihood that a correct one is included.

The set of all possible partial interpretations defines the problem-solving search space. The more alternative hypotheses generated, the larger the fraction of the space actually searched. Since each partial interpretation can give rise to multiple extensions, the possibility of a combinatoric explosion exists. At each step in the search, a subset of the existing partial interpretations is selected for extension; the extended partial interpretations that result then compete for selection with those previously generated. The selection of the subset of hypotheses to extend is called the *focus-of-control* (or *focus-of-attention*) problem. An integral part of effective focus-of-control is the problem-solving system's ability to focus quickly on information that constrains the search, in order to contain combinatoric explosions. This is called an *opportunistic* and *asynchronous* style of problem-solving. It can be implemented through the Hearsay-II formulation of the hypothesize-and-test paradigm, in which promising tentative decisions are made (despite incomplete information or knowledge), then re-evaluated later in the light of new information. Focus-of-control is discussed further below; it is also discussed more extensively in [Hayes-Roth 77a].

Three requirements must be met for the effective operation of this general approach to problem-solving:

- *Sufficiency of Knowledge*: The knowledge can generate some sequence of partial interpretations that culminates in a correct complete interpretation.
- *Sufficiency of Credibility Evaluation*: The credibility function rates the correct complete interpretation higher than any incorrect complete interpretation generated.
- *Sufficiency of Control Strategy*: The focus-of-control strategy can find a correct complete interpretation within the bounds of computing resources allocated to the task.

Increasing the constraint of knowledge, the discrimination power of the credibility evaluation, or the selectivity of the control strategy beyond that which is minimally sufficient to meet these criteria will, in general, decrease the amount of computing resources needed for the interpretation. Also, these three aspects of the problem-solving are not independent; within limits, the same performance can be achieved by trading-off the uncertainty resolving power of one aspect for that of another.

2.2. The Architecture

Figure 1 shows a simplified schematic of the centralized Hearsay-II architecture. The major data structures are the shared global database (called the *blackboard*), focus-of-control database, scheduling queues, and databases local to KSs.

The blackboard is partitioned into distinct information levels, each used to hold a different kind of representation of the problem space. The major units on the blackboard are the hypotheses.

- Relationships among hypotheses at different levels are represented by a graph structure. The

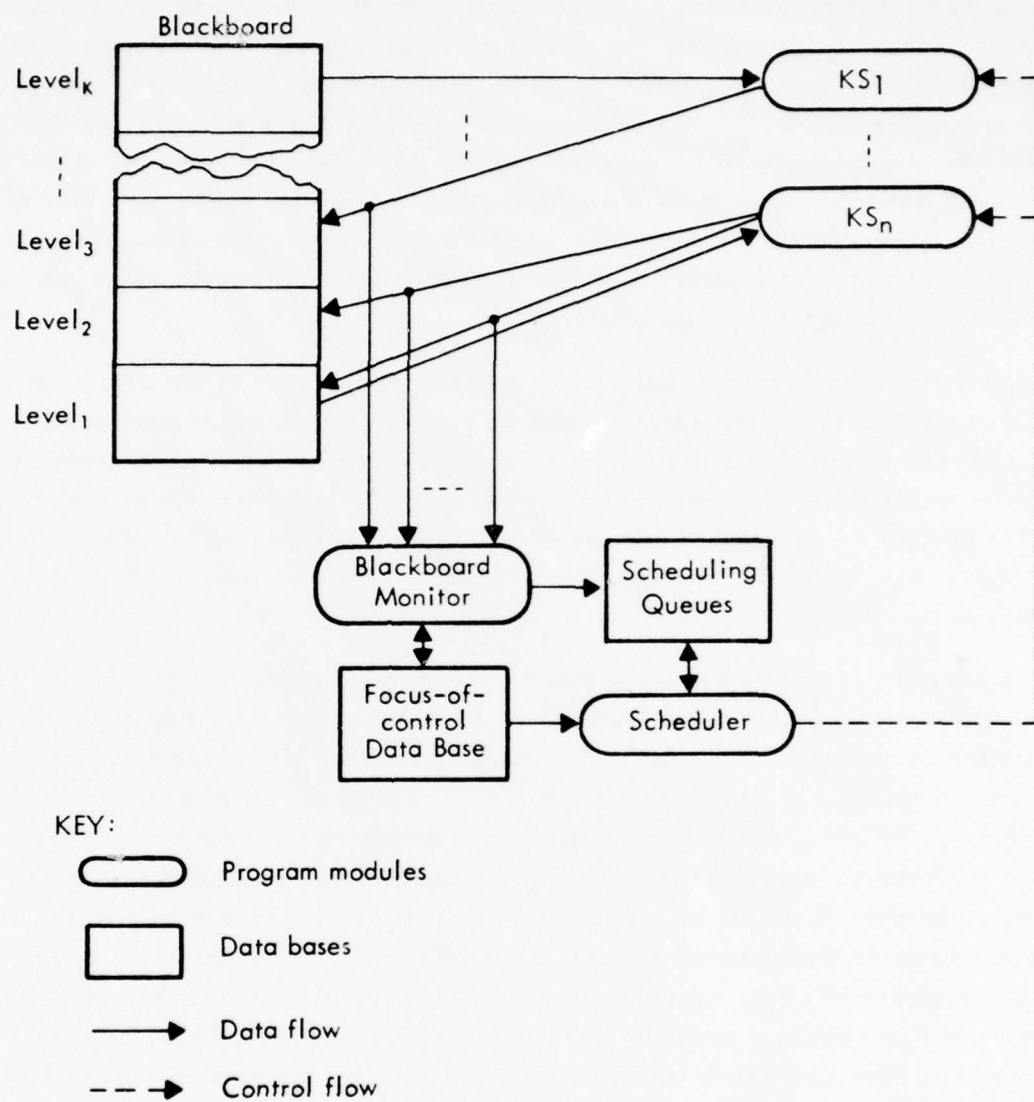


Figure 1: Schematic of the (centralized) Hearsay-II architecture.

sequence of levels on the blackboard forms a loose hierarchical structure in which the elements at each level can be described approximately as abstractions of elements at the next lower level. For example, in speech understanding an utterance can be represented as a signal, or as sequences of phones, syllables, words, phrases, or concepts; in image understanding, typical levels might include picture points, line segments, areas, surfaces, and objects; levels in an aircraft-tracking radar system might include signals, signal groups, vehicles, area maps, and overall area maps (see [Smith 78]). The set of possible hypotheses at a level forms a problem space for KSs operating at that level. A partial interpretation (i.e., a group of hypotheses) at one level can be used within the opportunistic strategy to constrain the search at another level. For example, a KS can create a phrase hypothesis as an abstraction of a sequence of word hypotheses. Similarly, another KS can use the phrase hypothesis to predict (i.e., constrain) the set of possible word hypotheses that might follow the phrase.

In order to implement the data-directed activation of KSs, each KS has two components: a pattern and an action. Whenever the pattern is matched by some hypothesis structure on the blackboard, an activation of the KS is created. If the KS activation is selected eventually by the scheduler, its action is executed in the context of the matched structure. For example, the pattern of a KS might be the creation of a new syllable hypothesis, and its action might be to use that syllable hypothesis and possibly other, adjacent syllable hypotheses to create new word hypotheses.

KS activity, and hence the search process, is managed by the scheduler using the focus-of-control database and the scheduling queues. At any point, the *scheduling queues* contain the pending KS activations. The scheduler calculates a priority for each waiting activation and selects for execution the one with the highest priority. The priority calculation attempts to estimate the impact of the information to be generated by an activation on the current state of the problem-solving. From the problem-solving viewpoint, the impact of some information is a measure of the degree to which it reduces the uncertainty of the interpretation, or, alternatively, the degree to which it reduces the number of competing interpretations. This measure changes as the problem-solving progresses; thus, the timeliness of creation of the information affects its impact. For example, if two pieces of information can lead to the same hypothesis, the creation of the first of them may have high impact, but the creation of the second will have little, other than adding confirmation to the hypothesis.

Several dimensions can be used to estimate the impact of information, including the following:

- The *credibility* of some information is a measure of the system's confidence in the information; the more credible the information, the higher its expected impact.
- The *scope* of some information is a measure of the amount of the total problem solution that it describes. It is related to the level of abstraction (e.g., in speech understanding, a word has larger scope than a syllable) and to the size (e.g., a two-second phrase has larger scope than a one-second phrase). The larger the scope, the greater the impact because a larger portion of the complete interpretation,

and hence more constraint, is specified.

- The *diagnosticity* of some information is a measure of how much competing information can be resolved by the information [Hayes-Roth 77b]. For example, if one part of the current partial solution has high credibility while another part has only low credibility, a moderately credible piece of information in the former area will have low diagnosticity, but a moderately credible piece in the latter area will have high diagnosticity and hence greater impact.

The *focus-of-control* database contains meta information about the state of the system's problem-solving activity. The meta-information is used to estimate the impact of information, based on its credibility, scope, and diagnosticity. Meta-information includes such things as the current best hypotheses on the blackboard and how much time has elapsed since these hypotheses were generated or combined with others. (This latter kind of information allows the system to recognize a state of stagnation in part of the problem-solving and then to cause the reappraisal of the impact of the current best hypotheses.) The focus-of-control database is updated by the blackboard monitor based on the generation and modification of hypotheses on the blackboard by KSs.

The blackboard monitor is also used to implement the data-directed activation of KSs. At system initialization, each KS declares hypothesis characteristics relevant to it. When an hypothesis is created or modified so as to match those characteristics, the blackboard monitor creates an activation record for the KS on that hypothesis and places it in the scheduling queues.

3. ISSUES IN DISTRIBUTING HEARSAY-II

Figure 2 presents a number of dimensions of decomposition of Hearsay-II for a distributed environment and several options for each dimension. From this table and the overview above, it can be seen that the characteristics of the Hearsay-II organization appear to make it suitable for a distribution along several dimensions:

- Information might be distributed: The blackboard database is multidimensional (with the information levels forming one dimension). Each KS activation generally accesses only a small, localized subspace within the blackboard.
- Processing might be distributed: Knowledge is encapsulated in KS modules that are largely independent, anonymous, and capable of asynchronous execution.
- Control might be distributed: KS activation is based on the generation and modification of hypotheses on the blackboard (data-directed control). To the extent that these hypotheses can be distributed, control of KS activation can also be distributed. The data-directed form of the hypothesize-and-test paradigm permits KSs to exchange partial results in a cooperative fashion.

Given these possibilities, it would appear that the Hearsay-II organization could be decomposed easily for a distributed environment so as to emulate efficiently and exactly the processing that occurs in the centralized version of the organization. In fact, a shared-memory multiprocessor implementation, using explicit synchronization techniques to maintain data integrity and distributed

** INFORMATION **

Distribution of the blackboard:

- The blackboard is distributed across the nodes with no duplication of information.
- The blackboard is distributed with possible duplication of information; synchronization techniques are used to insure consistency.
- The blackboard is distributed with possible duplications and inconsistencies.

Transmission of hypotheses:

- Hypotheses are not transmitted beyond the node in which they are created.
- Hypotheses may be transmitted directly to a subset of nodes.
- Hypotheses may be transmitted directly to all nodes.

In addition, the transmission and reception of hypotheses can be filtered based on characteristics of the hypotheses, e.g., type of hypothesis (information level), credibility rating, and location of the "event" the hypothesis describes.

** PROCESSING **

Distribution of KSs:

- Each node has only one KS.
- Each node has a subset of KSs. The selection might depend on factors such as the type of sensors at the node, the node's physical location, and the input/output characteristics of the KSs.
- Each node has all KSs.

Access to the blackboard by KSs:

- A KS activation can access only the blackboard in its local node.
- A KS activation can access blackboards in a subset of nodes.
- A KS activation can access blackboards in any node in the network.

** CONTROL **

Distribution of KS activation:

- A change to an hypothesis activates KSs only within the local node.
- A change activates KSs in a subset of nodes.
- A change activates KSs in any node.

Distribution of scheduling and focus-of-control:

- Each node does its own scheduling, based on local information.
- Each subset of nodes has a scheduler.
- A single, distributed database is used for scheduling.

Figure 2: Dimensions of decomposition for Hearsay-II.

along the processing and control dimensions, achieved significant parallelism -- a speedup factor of six [Fennell 77]. However, the following characteristics of Hearsay-II introduce a number of difficulties for such a straightforward emulation in a distributed environment:

- The scheduler, which requires a global view of the pending KS instantiations (scheduling queues) and the focus-of-control database, is centralized.
- The blackboard monitor, which updates the focus-of-control database and scheduling queues when a specific type of blackboard change occurs, is centralized.

The patterns of KS access to the blackboard overlap, prohibiting the construction of compartmentalized subspaces of the blackboard accessed exclusively by small groups of KSs.

Because there are many KS executions, each accessing the blackboard frequently, an extensive amount of interprocessor communication would be required to emulate exactly a centralized view of the blackboard, scheduling queues, and focus-of-control database. The dynamic information in these data structures controls the degree and nature of KS cooperation and is essential to the effective implementation of the hypothesize-and-test problem-solving strategy.

Given that the communication and synchronization costs of emulating perfectly the centralized views are too high, one is led to their approximation. The amount and range of internode communication can be reduced, leading to inconsistency and incompleteness of the local views and thus unnecessary, redundant, and incorrect processing. Experiments with the shared-memory multiprocessor Hearsay-II speech understanding system described above demonstrated that the system could operate in such an environment [Fennell 77]. In these experiments, the explicit synchronization was eliminated without degrading accuracy as measured at the end of processing, with an attendant increase in the speedup factor from six to fifteen because of the reduction in interprocess interference. The explanation for this phenomenon is that the asynchronous, data-directed control can apply knowledge to correct certain types of internal errors. Consider the normal activity sequence of a KS, which involves first examining the blackboard and then creating new hypotheses on the basis of the examined hypotheses. If the set of relevant hypotheses changes after the KS looks at them and before it modifies the blackboard, the modification would be inconsistent or incomplete with respect to the current state of the blackboard; however, because of the data directed nature of KS activation, the intervening changes will trigger the same KS to recalculate its modifications and perhaps generate new alternative hypotheses. In addition, other types of inconsistency can be resolved because additional KS processing will usually result in lower credibility ratings for an incorrect hypothesis and its extensions, whether the incorrect hypothesis resulted from a synchronization error, a mistake in the knowledge used by the KS, or from erroneous data. Thus, this self-correcting nature of information flow among KSs, created through the use of the incremental data-directed hypothesize-and-test paradigm, in many cases obviates the need for explicit use of synchronization.

The key issue is whether a distributed decomposition of a Hearsay-II-like system can be designed that can deal with the errors introduced by the approximate emulation well enough to

maintain satisfaction of the sufficiency criteria of Sec. 2.1. In the distributed system, internode communication becomes part of the "computing resources" that must be limited for effective system performance.

4. A NETWORK OF HEARSAY-II SYSTEMS

A primary goal of our decomposition design is to minimize internode communication relative to intranode processing. Because of this and the relatively fine granularity of KS activity within a Hearsay-II system, a node must be able to complete a number of KS executions in a self-directed way, i.e., without internode communication. Thus, each node in the network must contain KSs, a scheduler and focus-of-control database for selecting the next KS activation to execute at each step, a blackboard for KS communication, and a blackboard monitor for KS activation. Therefore, each node is an architecturally complete Hearsay-II system.

There are dual points from which to view the distribution of the dynamic information (i.e., partial interpretations and meta-information) in the network:

- A virtual global database represents all the system's information; the local databases at each node contain the node's partial view of the virtual global database, perhaps with some inconsistencies (because of limited internode communication and synchronization).
- Each node has its own databases; the union of these across all the nodes, with any inconsistencies, represents the total system interpretation.

The first viewpoint corresponds to the way most distributed computing systems are considered -- a centralized system is *decomposed*, with each piece (node) in the decomposition viewed as a part of the whole system. From the second viewpoint, the distributed system is *synthesized* from systems operating at each node. The second approach shifts the view from that of a system distributed over a network to that of a network of cooperating systems, each able to perform significant local, self-directed processing. Another way of distinguishing these viewpoints is that the first considers each node from the context of the whole system, while the second considers the system from the context of the individual node. When considering any particular design choice, one or the other of these viewpoints might be more appropriate.¹ From either viewpoint, the major design decisions are the selection and focusing of knowledge sources at each node and the choice of mechanisms and policies for internode communication to permit effective cooperative problem-solving. We will now describe some possibilities for each of these areas.

¹The general basis of these notions is expressed in the theory of Nearly Decomposable Systems devised by Simon [Simon 62] to describe complex organizational structures. The term "nearly decomposable" emphasizes the fact that systems can be decomposed naturally into clusters that have a high degree of intracuster activity and a lower degree of intercluster interaction. These dual views follow logically from the recursive nature of this hierarchical structure.

4.1. Intranode Considerations -- Selection and Focusing of KSs

Intranode processing can be maximized relative to internode communication if KS activity is such that the inputs needed by KS actions are available on the node's blackboard. Thus, the selection of KSs for each node and the focusing of their activity on particular portions of the problem greatly affects this goal.

The blackboard in a Hearsay-II system is described along several dimensions. One of these is *information level*; this dimension has discrete points, each corresponding to a different way of representing the situation being interpreted. A KS typically works with a small number of information levels by noticing one or more hypotheses (called the "stimulus") at one or two levels and by creating new hypotheses or modifying existing ones (the KS's "response") at one or two levels. For a collection of KSs to be connected across levels, then, it must be that any level used by some KS as its stimulus is used by some KS as its response. There are also KSs that are transducers between the system (i.e., the blackboard) and the external world. For the purposes of this discussion, we will think of an input transducer as having no blackboard stimulus and an output transducer as having no blackboard response. In a network of Hearsay-II systems, if a particular node has a KS which is level-disconnected on its stimulus or response side, that node is forced to communicate with other nodes to supply the missing stimulus or to provide a use for the "extra" response. Since a primary goal is to maximize intranode processing relative to internode communication, the selection of KSs for each node should maximize the level-connectivity. Likewise, transducer KSs should be selected for their appropriateness to the particular types of sensors (and effectors) at the node.

In addition to the information level, there is an orthogonal dimension (or set of dimensions) for locating hypotheses in the blackboard -- this is the *location* of the event which the hypothesis describes. For signal interpretation tasks this usually represents a physical location. In speech understanding, for example, most hypotheses (phones, syllables, words, phrases, etc.) can be located as segments on the dimension of time within the utterance. For image understanding, objects (at any of the levels) can be located in the two or three dimensions of the image space. For radar tracking of aircraft, signals and objects can be located in the three-dimensional world. In general, hypotheses closer in the location dimension are more likely to be relevant to each other and to be needed jointly for further KS activity. For example, a word hypothesis is likely to be created from adjacent syllable hypotheses, an object is likely to be created from surfaces near each other, and a signal group from signals detected nearby. Thus, a node should attempt to acquire for its local blackboard all of the hypotheses at a given level within a contiguous segment in the location dimension(s).

All levels in the system taken together with the full extent of the location dimension(s) define a node's largest possible scope. The term *area-of-interest* will be used to denote, for each node, that portion of the maximum scope representable within the node's local blackboard.

The levels in the area-of-interest are the union of the stimulus and response levels of the KSs

in the node -- any other levels would be useless to the node.¹ A node's area-of-interest at the information level(s) to which the sensory data is transduced should cover in the location dimensions at least the area covered by the node's sensors; otherwise, some of the sensory data would be lost, since the only direct action the transducer KS can take is to create hypotheses on the local blackboard about the data.² At the other levels, the location segment should probably include at least the projection of the location segment at the transduction level, since it is reasonable to create higher-level hypotheses about the locations covered by the node's sensors. In addition, the location segment should also likely be extended somewhat beyond the range of the local sensors; this is to allow the node to acquire information from neighboring nodes to use as context for KS processing. Finally, this context extension should probably be larger at higher information levels, because the size of hypotheses (i.e., their length in the location dimension(s)) tend to be larger at the higher levels; e.g., words are usually bigger than syllables, objects are usually bigger than surfaces, and area maps larger than aircraft.

As an aid to understanding the notion of area-of-interest, let us consider a simple example of bottom-up processing at a single node of a network operating in a one-dimensional location space. The node has three information levels, labeled L1, L2, and L3, and two knowledge sources, KS1 and KS2 (see Fig. 3). Hypotheses on L1 are uniformly one unit long in the location dimension and are contiguous and non-overlapping. The sensor associated with the node produces a single hypothesis on L1, called H1, at location 50.³ Knowledge source KS1 in the node can take three contiguous hypotheses on L1 -- call them H2, H1, and H3 -- and produce H4 as an abstraction of them on L2. Likewise, knowledge source KS2 produces hypotheses on L3 from triples of hypotheses on L2.

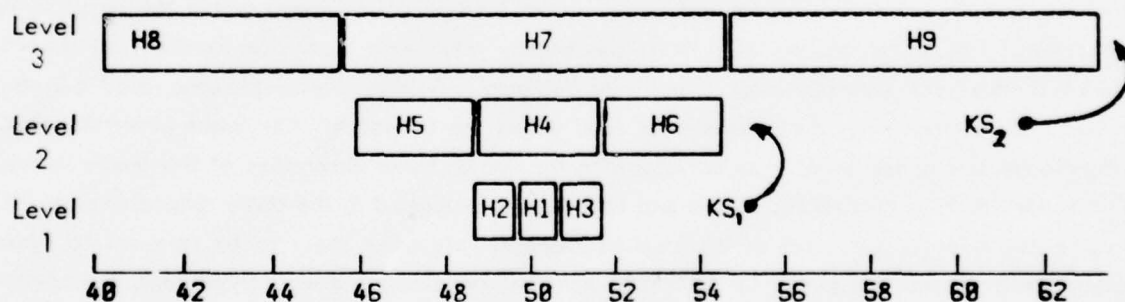


Figure 3: Simple example of area-of-interest.

¹In Sec 4.4.2, we will show one use for representing hypotheses which cannot be processed by local KSs, in particular, for allowing a node to act as a store-and-forward message handler.

²Of course, the transducer could use the sensory information to modify hypotheses about adjacent areas, but this would represent the sensory information only indirectly.

³In general, multiple, alternative, competing hypotheses could be produced throughout this example, but we will not consider them here.

In order for KS1 to operate, the node must receive hypotheses H2 and H3 as messages from some other nodes, because its local sensor can generate only H1. Likewise, for KS2 to operate, the H5 and H6 hypotheses must be received on L2. The scope required to be representable on L2 is larger than on L1. If processing were to continue similarly above L3, L3's scope would have to be larger still. Thus, the location dimension of the area-of-interest expands at higher levels. The lateral communication (e.g., H2 and H3, and H5 and H6) forms a context for processing and provides a connectivity in the location dimension (*lateral connectivity*), similar to the connectivity in the information-level dimension.

The particular scope of the area-of-interest is dependent on the information required by the KSs. In this simple example, KS2 is able to create hypotheses on L3 based solely on the information on L2. If KS2 required information about a L2 hypothesis that is not represented in the abstraction on L2, it will want to look at the L1 substructure of the hypothesis. If the information needed is about H4, KS2 can access it on the node's blackboard directly, looking at hypotheses H2, H1, and H3. If, however, KS2 needs to look at the substructure of H5 or H6, there is a problem because the L1 representations of those hypotheses are not on the node's blackboard. One solution is to have KS2 do the best it can without the information, thus requiring no additional internode communication but introducing additional uncertainty in the problem-solving. Another solution to this problem is to extend the node's area-of-interest on L1 in order to represent the needed information. This extension can be handled in several ways:

- A priori analysis of KS2 indicates that the L1 information is likely to be needed. Thus, the scope of the node's area-of-interest on L1 is permanently specified to be 46-54, and the node gathers all L1 information that it receives. If the needed information is less than the full scope, the expansion of the area can be limited. For example, if information about just boundaries of the L2 hypotheses is needed, the scope could be specified as 48-52, rather than 46-54.
- Each node that transmits L2 hypotheses knows that some of the corresponding L1 information is likely to be needed; they therefore transmit the relevant L1 information whenever they transmit an L2 hypothesis. Thus, the scope of the receiving node's area-of-interest on L1 dynamically expands in response to the reception of L2 hypotheses.
- When KS2 discovers the need for the L1 information, it expands the scope of the node's area-of-interest so that it is capable of representing the needed information if it is received. KS2 then processes as best it can without the information, perhaps creating no L3 hypothesis. If the needed L1 information is subsequently received, KS2 can be retriggered to re-evaluate the earlier action and perform corrective modification if needed.¹

The suggestions here for defining the area-of-interest of a node are only one possible set of

¹ There are a variety of approaches for acquiring the needed information which involve more explicit communication among nodes. For example, attached to each transmitted hypothesis is the name of the sender so that later point-to-point communication might be established. Even though the basic approach to internode communication developed here is based on a more implicit communication approach (similar to the way KSs communicate through the blackboard), we briefly discuss some of these more explicit approaches in Sec. 4.4.2.

guidelines; others could be used. The area can also be adjusted dynamically to adapt to changing conditions, such as movements of the node or its sensors or changes in demands on the node's processing or memory capacity. What is important is that each node has an area-of-interest that defines its blackboard and thereby puts bounds on the area in which local processing can occur and on what information is important for it to receive. As suggested by the example in this section, the particular sections of the area-of-interest from which information needs to be transmitted and received are task-specific, depending upon the specific requirements of the KSs and their selection and focusing in the network.

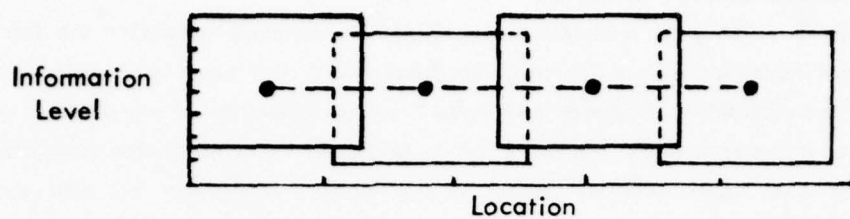
4.2. Network Configurations

Within the guidelines developed so far, a variety of organizational structures can be implemented in the network, depending on the selection and focusing of KSs in each node. For example, if all nodes contain the same set of KSs and levels, the network structure is "flat" and information flow is essentially lateral. This is the simple structure of the system used for the experiments described in the rest of the report. Figure 4a represents such a flat configuration.

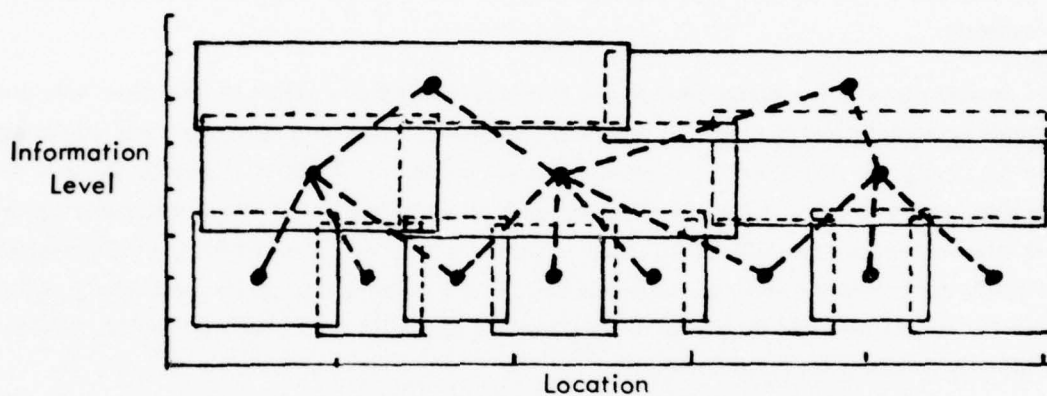
More complex processing organizations occur where there is a non-uniform distribution of KSs and levels across the nodes. Figure 4b shows an overlapping hierarchical structure. Figure 4c shows the implementation of what is called a "matrix" configuration in organizational structuring (see, for example, [Galbraith 73]). In this configuration, each of a set of general-purpose nodes (at the higher levels) makes use of information from lower-level specialists.

Figure 4 shows simplified schematics of the configurations, indicating the levels in each node's area-of-interest, its approximate position in a one-dimensional location scheme, and the internode communication paths. This figure does not indicate the intensity of communication, from what sections in an area-of-interest information is being transmitted, whether the paths are bidirectional, or the actual shape of the area-of-interest -- varying these parameters leads to greater varieties of network configurations.

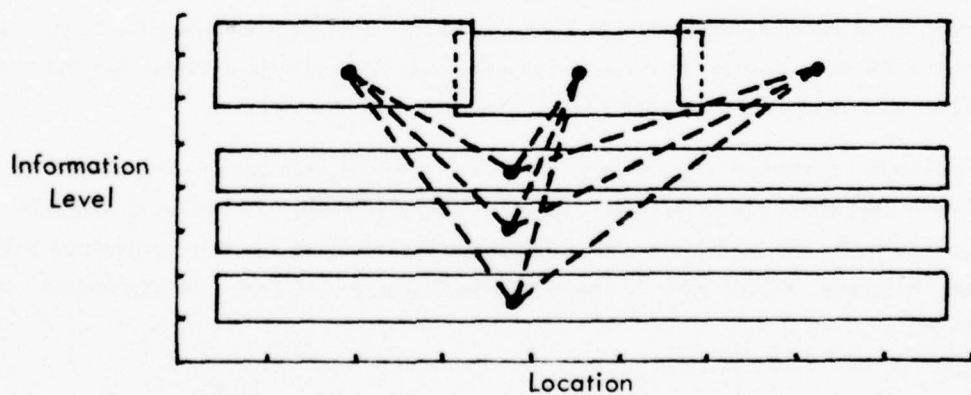
The emphasis throughout this report is on the flow of information among nodes, with each node cooperating but having control autonomy. Within this paradigm, various control relationships can be synthesized implicitly by establishing particular information flow paths, resulting in appropriate data-directed activity of nodes. A more explicit implementation of control relationships can be integrated with information flow through the use of a mechanism in Hearsay-II called a *processing goal* [Lesser 77]. This is an information structure a KS creates on the blackboard as an active request for information of a particular type. KSs which can produce such information may then respond to the goal in the same way they would to the creation of a relevant hypothesis. When a goal is transmitted between nodes, as with any other hypothesis, the same kind of request-response activity can occur. A more extended version of this notion which involves a two-way dialogue is the central idea in the contract net formalism for resource allocation in a distributed environment [Smith 78].



a: Schematic of a "flat" configuration.



b: Schematic of an overlapping hierarchical configuration.



c: Schematic of a matrix configuration.

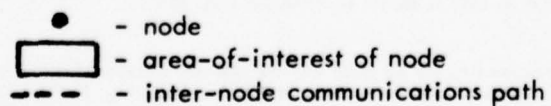


Figure 4: Schematics of some network configurations.

4.3. Internode Communication -- Mechanism

In a Hearsay-II system, all inter-KS communication is handled indirectly via the creation, modification, and inspection of hypotheses on the blackboard. This same mechanism may be used for internode communication. Consider a Hearsay-II system operating at one node in a network, with its area-of-interest defining the scope of its blackboard and hence the possible areas of attention of its KSs. Now consider adding to that node a transducer KS with access to a communication medium (e.g., packet radio) for receiving messages from other nodes describing their hypotheses; if this *RECEIVE KS* modifies its node's blackboard to reflect those messages, other KSs in the node can use this information. Likewise, a *TRANSMIT KS* can select hypotheses on the blackboard and transmit them for reception by other nodes. Figure 5 shows a network of such systems.

The decision to use the blackboard as the sole means of KS interaction in Hearsay-II was made to provide uniformity and to keep KSs relatively independent of each other. The same advantages accrue by using the blackboard for internode communication. A KS is triggered by and uses information on the blackboard independent of what other KS created it; thus, information placed on the blackboard by the *RECEIVE KS* is automatically usable by the other KSs, indistinguishably from locally generated information. Likewise, each KS posts its results on the blackboard without concern for what other KSs might use it; thus, the information to be transmitted by the *TRANSMIT KS* is already available on the blackboard.

A node could transmit, in addition to hypotheses, waiting KS activation records from its scheduling queues, in order for them to be executed at another node. If a node receiving such an activation record has both the KS and blackboard data needed for executing the activation, the data-directed nature of KS activation would have already created an equivalent activation locally. If either the KS or data are not present, the activation could not be executed by the receiving node. Thus, it is redundant or useless to share the scheduling queues.¹

KSs in Hearsay-II interact asynchronously. That is, a KS triggers whenever an event occurs of interest to it and, when executed, makes use of whatever relevant information is available on the blackboard to make the best statement it can about the situation. Such asynchronous intranode operation naturally allows KSs to handle asynchronous internode communication without modification.

4.4. Internode Communication -- Policies

The ability to run asynchronously eliminates the need for communication costs of synchronization and simplifies the interaction mechanisms. There is still a need to reduce the amount of internode

¹We are assuming here that the environment for KS execution (i.e., the KS itself and the relevant blackboard data) is not transmitted. One could consider transmitting such information with KS activations for internode load-balancing. One could also consider transmitting activations and the node's priority evaluation of them in order to influence the scheduling decisions of other nodes.

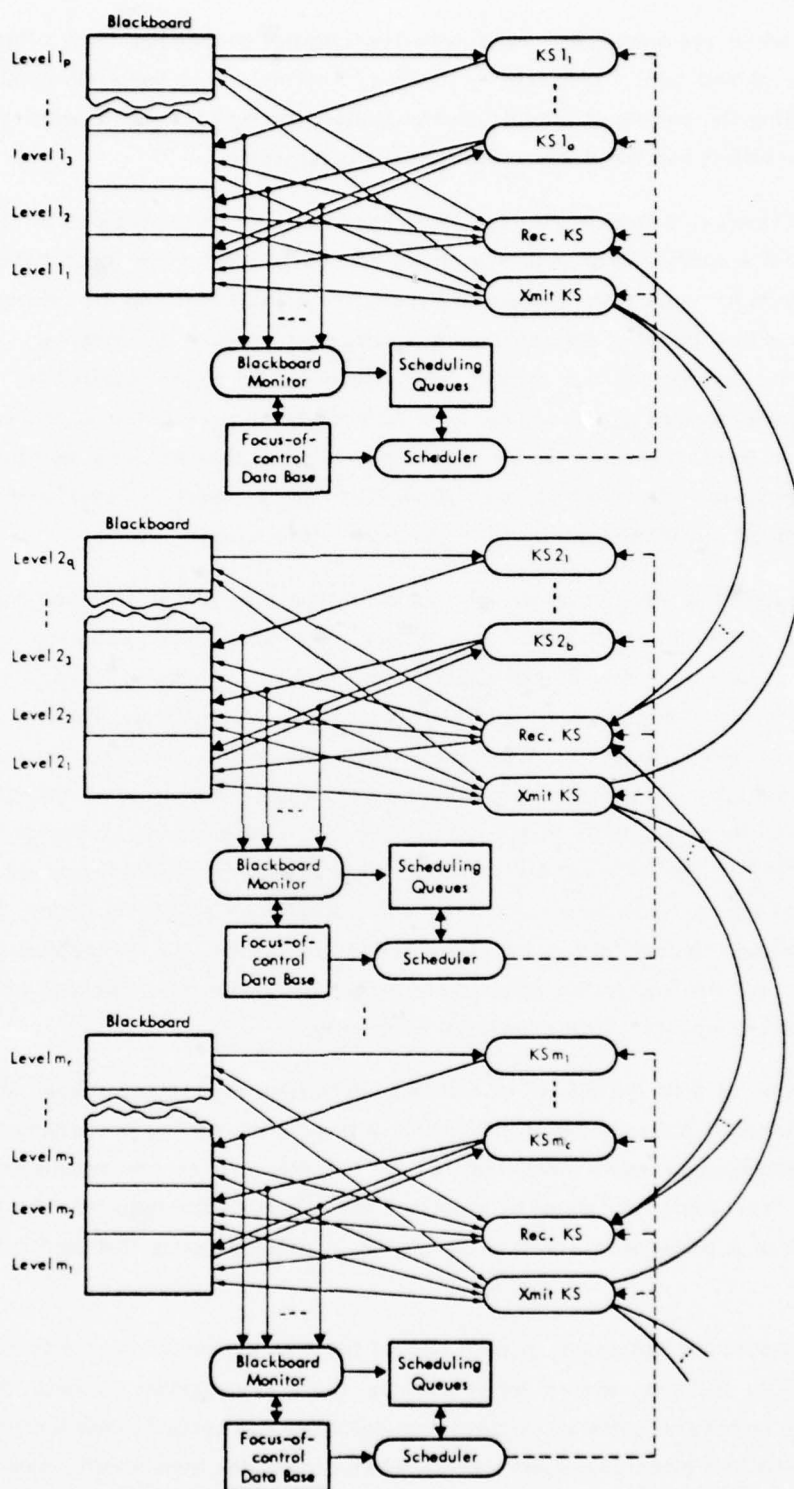


Figure 5: Schematic of a network of Hearsay-II systems.

communication while providing each node with the information needed from other nodes (i.e., guaranteeing level and lateral connectivity of KS processing). Internode communication can be reduced by limiting the amount of information transmitted, the set of nodes to which any particular message is transmitted, and the distance the message is transmitted.

A centralized Hearsay-II system must limit the number of hypotheses created on its blackboard, in order to avoid a combinatorial explosion of KS activity in reaction to these hypotheses. The primary mechanism for limiting the number of hypotheses is the structuring of a KS as a generator function. One activation of a KS can create a few most credible hypotheses. Stagnation of progress of those hypotheses can trigger new activations to create alternative, less credible hypotheses. Asynchronous KS interaction, as described above, permits the additional hypotheses to be exploited in the same manner as the original hypotheses. Similarly, in a distributed system a node does not need to transmit all its information; rather it can select its "best" and subsequently respond to the need for additional information by transmitting more.

The transmission of a piece of information is worthwhile only if it is received by a node that finds it relevant. At one extreme, each transmission could go to all nodes and each node would be responsible for selecting relevant information from its received communications -- this global broadcast scheme would require relatively high bandwidth. Alternatively, the transmitting node could know which other nodes might be interested in the information and thereby direct the communication explicitly. The cost of maintaining such a complete distributed knowledge of what is relevant to each node would be high, especially since the information changes as the problem-solving progresses. The scheme we consider here is a local transmission based on local knowledge of relevance. Each message is transmitted to a few neighboring nodes. When a node receives information relevant to it, it incorporates the information into its problem-solving state. This action may in turn trigger the node to retransmit the information (perhaps modified by its knowledge), on the basis of its local knowledge of relevance.

The transmission of a limited subset of a node's information to a limited subset of other nodes leads to an incremental transmission of information with problem-solving processing at each step, similar to the relaxation paradigm [Rosenfeld 76]. This transmission scheme results in what can be thought of as a "spreading excitation" of important news through the network. As in relaxation, the propagation of a piece of information dies out as it reaches nodes that find it irrelevant or unimportant.

Local knowledge-based processing at each step of the transmission can serve to correct errors in the information, including errors introduced by the communication process itself. Since communication is incremental, this error-correction capability can serve to limit the propagation of errors, as opposed to a global broadcast scheme, which propagates them widely. One drawback of the incremental transmission strategy is the increase in the time needed to communicate important information across the net, because each local step adds some delay. However, a node's information is generally most directly relevant to nodes nearby, and the information contained in

these neighboring nodes is generally more constraining (i.e., error-correcting) than that of nodes farther away. Another drawback is the possibility that the transmission of important information will die out because the local measures of importance may be incorrect. This danger is reduced because of the correlation between the proximity of nodes and their measures of relevance. It can be reduced farther by increasing the richness of connectivity of the internode communication paths, at the cost of additional communication.

In order for one node to have information relevant to another, their areas-of-interest must overlap, since each node's area-of-interest defines what is of interest to it. Thus the selection of areas-of-interest also constrains the potential internode communication patterns. The criteria for selecting the area-of-interest given in Sec. 4.1 led us to place the center of a node's area at the location of the node's sensors. Thus, geographically proximate nodes -- i.e., those with sensors proximate in the location dimension -- have more overlap in their areas than nodes which are further apart, and therefore have more to communicate.

The incremental communication strategy is also more economical, since communication between nodes is generally less costly the closer they are. This is certainly true if the communication medium is hard-wired lines. It is also true for radio; in fact, as the distance that messages need travel is reduced, the power requirement is reduced (and with it the cost of hardware). Also, the same broadcast channel can be used simultaneously in different parts of the network without as much interference.

In order to implement such an incremental communication system, three policies must be specified:

- the RECEIVE KS's integration of received information onto the blackboard,
- the TRANSMIT KS's selection of information to transmit, and
- the determination of which nodes will communicate.

At the heart of these different policies are measures of the relevance (i.e., expected impact) of information for the processing at individual nodes. As described in Sec. 2.2, estimating impact is an important part of the focus-of-control issue for the centralized problem-solving system and meta-information (called the "focus-of-control database") plays a key role in this estimation. Because this meta-information attempts to measure the current state of progress in the problem-solving system, it requires a global view of the problem-solving database (the blackboard). In attempting to develop mechanisms to distribute the meta-information among the nodes, there is a tradeoff between the accuracy and scope of this information on one hand and the cost of acquiring it on the other. The more accurate and globally representative this meta-information is, the better the estimate of the relevance of local processing to other nodes. Better estimation leads to lower transmission bandwidth requirements, less redundant processing, and more responsiveness of the system to new, important information. However, the cost of acquiring the more accurate meta-information has its own attendant bandwidth and processing costs that can possibly outweigh the advantages of better local estimates. This tradeoff is classic to all resource-allocation problems, i.e., the cost of doing the allocation (in terms of processing and information acquisition necessary to support it) versus the resources saved by doing it.

4.4.1. The Basic Policy

The basic policy for communication to be considered is for a node

- to accept any received information that is representable within its area-of-interest and to integrate that information onto its blackboard as if it were generated by local KSs (and hence update its meta-information accordingly),
- to select for transmission those hypotheses whose estimated impact is highest and which have not been previously transmitted, and
- to broadcast them to all nodes that can receive the communication directly.

This policy is simple in that communication is not directed to specific receiving nodes, no distinction is made between locally generated and externally received hypotheses, and the mechanism used to control local activity is also used to select hypotheses to be transmitted.

This policy leads to the same kind of generator behavior that is produced in the local KS activity: High-impact hypotheses (locally decided) are transmitted initially. If, after a time, no higher-impact hypotheses arrive on the node's blackboard (either generated locally or received from some other node) that subsume or compete with these transmitted hypotheses, the stagnation mechanism will cause other, previously lower-rated hypotheses now to be rated high-impact and hence transmitted.

Since a node's meta-information is strongly dependent on those hypotheses that are judged high-impact, and since it is those hypotheses which are transmitted, a receiving node, by incorporating those hypotheses and modifying its meta-information accordingly, will implicitly incorporate a large part of the sender's relevant meta-information. Thus, the meta-information will also be "relaxed" across the network.

We will now discuss some variants of this basic policy. These respond to particular characteristics of the problem-solving task and the communication channels.

4.4.2. Variants

If the reliability of the problem-solving processing is such that most hypotheses of small scope are incorrect and if most of the small-scope hypotheses can be refuted by additional processing within the creating node, then it may be better to transmit only hypotheses for which the node has exhausted all of its possible local processing and which come through that processing with a high impact measure. This strategy, called *locally complete*, can 1) reduce the communication bandwidth needed, since fewer hypotheses need to be sent (just those that survive unrefuted), 2) reduce the processing requirements of the receiving nodes, since they will have fewer hypotheses to incorporate and judge, 3) avoid redundant communication in the case that two nodes have a large area-of-interest overlap, and 4) increase the relevance of transmitted hypotheses because their scopes are larger (due to the additional processing) and thus more likely to overlap

areas-of-interest of other nodes. The potential disadvantage is the loss of timeliness -- the earlier transmission might provide significant constraint for the receiving node.

A technique we call *murmuring* can be used to improve the reliability of communication. In this technique, a node retransmits high-impact hypotheses. A simple approach is to murmur periodically, independent of other communication. A more efficient approach is to murmur high-impact hypotheses unless the node receives or generates higher-impact hypotheses. The stagnation measures (see Sec. 2.2) can be used to implement this strategy. Murmuring is a knowledge-based technique that can be used to correct for lost communications due to intermittent channel or node failures and to bring up-to-date new or moving nodes, thereby gaining some measure of dynamic network configuration. This mechanism has the advantage of preserving anonymity of communication and requires no explicit hand-shaking or acknowledgment.

The mechanisms described so far involve the acquisition by each node of a model of the processing state of other nodes implicitly through the problem-solving information received by the node. Such implicit mechanisms are simple, but may not be efficient enough for some cases. For example, the assumption that nodes which can communicate directly have overlapping areas-of-interest is needed to guarantee that relevant and needed information is propagated throughout the network; if, however, there are discontinuities or insufficient redundancy in these overlaps, a more explicit mechanism is needed to guarantee a rich enough connectivity to handle the problem-solving.

One way to handle such problems is for a node to transmit a description of its area-of-interest, explicitly indicating what kinds of information it needs and what kinds it can produce, i.e., its *input/output (I/O) characteristics*. Each node receiving this message responds with a reply containing its I/O characteristics. If the initiating node is unsatisfied with the richness of the neighborhood connectivity implied by the responses, it can transmit another message, indicating which of its I/O requirements are not sufficiently satisfied and requesting its neighbors to ask their neighbors, in turn, to fulfill them. The initiating node can continue expanding the area of its request until all of its requirements are met or until it decides to give up. Subsequently, the intermediate neighbors will act as store-and-forward message processors supporting the desired connectivity. This provides a mechanism for generating explicit communication paths between nodes that have no direct communication capabilities. This may be necessary for some of the more complex network configurations, e.g., as in Figure 4c, in which overlapping areas-of-interest do not necessarily imply the geographic proximity of the nodes.

This process can be viewed as the dynamic increase of the area-of-interest of each intermediate node so that it can now accept the kind of information that it is being requested to forward. Even though the intermediate node might do no local problem-solving processing of this information, once it has accepted it, the normal criteria for transmission can handle the forwarding function.

Modification of a node's area-of-interest in response to explicit meta-information can also be used for resource allocation. For example, if a node has completed all possible processing within its area-of-interest and does not expect any new tasks to appear within that area-of-interest for some time, it may be worthwhile for it to advertise for new work, using a mechanism similar to that used for insuring connectivity. On the other hand, if a node finds that the demands on its local processing power are too great, it might shrink its area-of-interest, thereby reducing the domain of its activity. If there is sufficient overlap of areas-of-interest, this results in just a reduction of redundancy; if the overlap is not sufficient, a renegotiation, using the I/O characteristics, is needed to assure coverage of the whole problem. An exploration of these ideas will appear in [Lesser 79].

It may be useful to transmit other meta-information with hypotheses: for example, the name and location of the sending node, the time the hypothesis was generated, the amount of computing effort expended on the hypothesis, and the number of nodes that previously processed the hypothesis. The receiving node can augment its meta-information with this information.

Figure 6 summarizes the design decisions we have made along each of the dimensions of Fig. 2.

**** INFORMATION ****

Distribution of the blackboard:

- The scope of a node's local blackboard defines its area-of-interest.

Transmission of hypotheses:

- A node transmits hypotheses to a local subset of nodes.

**** PROCESSING ****

Distribution of KSs:

- Each node has a subset of KSs.

Access to the blackboard by KSs:

- A KS activation can access only the blackboard in its local node.

**** CONTROL ****

Distribution of KS activation:

- A change to an hypothesis activates KSs only within the local node.

Distribution of scheduling and focus-of-control:

- Each node does its own scheduling, based on local information.

Figure 6: Design decisions for a network of Hearsay-II systems.

5. THE EXPERIMENT

An experiment was performed to determine how the problem-solving behavior of such a network of Hearsay-II systems compares to a centralized system. The aspects of behavior studied include the accuracy of the interpretation, time required, amount of internode communication, and robustness in the face of communication errors. This experiment was a simulation only in part, since it used an actual interpretation system analyzing real data, i.e., the Hearsay-II speech understanding system [Erman 79].

5.1. Simulating a Network

The simulation aspects of the experiment involved emulating a distributed network of nodes with a broadcast communication structure. This was accomplished by developing a multi-job coordination facility for the Decsystem TOPS-10 operating system. This facility coordinates communication and concurrency among a collection of independent jobs, each running a Hearsay-II speech understanding system. The network communication structure is simulated by a shared file that holds a record of each transmission in the network and additional information, such as when and by which node it was generated and which nodes have read it. All jobs can access this file through an internode communication handler added to the basic Hearsay-II system. The simulation of concurrency among the jobs is accomplished by keeping the jobs' clock-times in step; each time a job makes a request to transmit or receive internode communication, it is suspended if its local processor time is no longer the smallest. In this way, the simulation of concurrency is event-driven rather than sampled; this permits accurate measurement and comparison of concurrent events across simulated nodes.

5.2. Selection of KSs and Areas-of-interest

A major design decision in the decomposition of a system is the selection and focusing of KS processing at each node. In the case of the Hearsay-II speech understanding system, the decision was to allocate all the KSs to each node. The area-of-interest for each node has all the information levels, but is restricted to a statically assigned segment of the location dimension, i.e., to a segment of the speech signal. Two aspects of the particular blackboard structure and KS configuration of the Hearsay-II system used in this experiment motivate this design.

The first aspect concerns how hypotheses are located on the blackboard. The information levels of the Hearsay-II speech understanding system are shown in Fig. 7. The position of an hypothesis on the location dimension is defined by its time segment within the spoken utterance. For example, a hypothesis might be that the word "today" occurred at the word level from millisecond 100 to millisecond 600 in the utterance. One can think of each node as having a microphone sensor which acquires its input from a segment of the utterance. As discussed in Sec. 4.1, it is natural to define a node's area-of-interest as being centered, in the location dimension, over its sensor's area. *Thus we are led to a one-dimensional network with each node listening to some portion of the utterance and with the portions overlapping.*

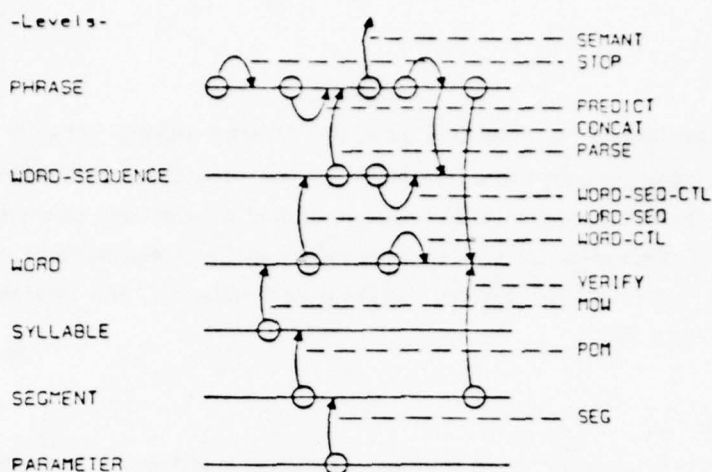


Figure 7: Levels and knowledge sources of the speech understanding system. The levels are indicated by solid horizontal lines and are labeled at the left. KSs are indicated by vertical arcs with the circled end indicating the level of its input and the pointed end indicating the level of its output. The name of a KS is connected to its arc by a dashed horizontal line.

The second aspect concerns the propagation of information across levels of the blackboard. KS processing in this version of the Hearsay-II speech system (see Figs. 7 and 8) is bottom-up and pipelined (without feedback) until the word-level is reached; i.e., all segments are created, then all syllables, then a selection of words. Additionally, the context of hypotheses required for KSs operating at these levels is highly localized in terms of position within the utterance -- i.e., in the location dimension. Thus, by choosing the areas-of-interest to have sufficient size and overlap in the location dimension, it is possible to guarantee that all bottom-up processing to the word level can be accomplished with no internode communication -- i.e., there is no need for communication to maintain lateral connectivity for this processing -- at the cost of possible redundant processing. The "sufficient" size and overlap criteria must be such that all possible valid hypotheses at these levels can be hypothesized because their time regions lie totally within at least one node.

Above the word level, a more incremental, data-directed form of processing occurs in which the context of hypotheses required for KS processing cannot be localized in the time dimension. In particular, phrase hypotheses must be transmitted among nodes.

Additionally, KS processing at the phrase level often requires the detailed characteristics of the underlying word support for the phrase abstractions. As discussed in the example in Sec. 4.4.1, there are a number of possible approaches to providing the appropriate information to a node. The approach taken here is to transmit explicitly with each phrase hypothesis the name, rating, and time-region characteristics of each word contained in its underlying word support. However, there is still a limitation on the scope of a node's area-of-interest at the phrase level since local

Signal Acquisition, Parameter Extraction, Segmentation, & Labeling:

- SEG: The signal is digitized, a set of parameters is created, and a labeled segmentation is produced.

Word Spotting:

- POM: From the segments, syllable hypotheses are created.
- MOW: From the syllables, word hypotheses are created.
- WORD-CTL: This "word-control" KS controls the number of word hypotheses that MOW creates.

Phrase-Island Generation:

- WORD-SEQ: From the word hypotheses and some grammatical knowledge, word sequences are created that represent potential phrases.
- WORD-SEQ-CTL: This controls the number of hypotheses that WORD-SEQ creates.
- PARSE: Given a word-sequence, this KS attempts to parse it. If successful, a phrase hypothesis is created.

Phrase Extending:

- PREDICT: Given a phrase, this KS predicts all possible words that might syntactically precede or follow.
- VERIFY: Given a phrase and predicted word, this KS rates how well the segments of the utterance support the word's existence adjacent to the phrase.
- CONCAT: Given a phrase and verified adjacent word, this KS creates a new phrase hypothesis of the extended phrase.

Rating, Halting, and Interpretation:

- RPOLE: This KS generates a rating for each new or modified hypothesis, using information placed on the hypothesis by other KSs.
- STOP: This KS decides when to halt processing (on the basis of finding a complete sentence with a sufficiently high rating or on the system's expending a prespecified amount of resources) and selects a phrase hypothesis (or set of phrase hypotheses) as the output.
- SEMANT: Given the selection output, this KS generates an interpretation in an unambiguous form for interaction with the information-retrieval system to which the user is speaking.

Figure 8: Functional description of the speech understanding KSs.

KS processing at that level can merge disjoint phrase hypotheses into an enlarged phrase hypothesis only if their juncture at the segment level is contained in the area-of-interest of the node. This requirement must be met in order for the KSs to ascertain that particular acoustic phenomena occur at the juncture. This implies that a received phrase hypothesis that does not overlap the node's area-of-interest at the segment level should be discarded.

5.3. Communication Strategy

The previous section defines the type of information to transmit (phrase hypotheses and their underlying word support) as well as the policy for its reception (i.e., ignore all received hypotheses that do not overlap the area-of-interest). What remain to be described of the communications strategy are the mechanisms for determining which phrase hypotheses should be transferred and to which nodes they should be sent. Three policies were explored for selecting hypotheses to be transmitted.

The first policy, called "full transmission", is to have no selection criteria and to transmit each phrase hypothesis as soon as it is created. This policy provides a benchmark for the other policies and simulates a nonsynchronized, centralized blackboard at the phrase level.

The second policy, called "dynamic thresholding", corresponds to the basic policy presented in Sec. 4.4.1 and uses the local focus-of-control database as a basis for evaluating the importance of a locally generated phrase hypothesis. The focus-of-control database keeps track of the best phrase hypothesis created (or received) for each time area of the utterance. The criterion for "best" hypothesis is constantly re-evaluated on the basis of whether a hypothesis has been successfully extended into an enlarged hypothesis -- if not, its rating is decreased, possibly resulting in the choice of another hypothesis to replace it as the best in the area. The criterion for transmission using this policy is straightforward: transmit an hypothesis when it becomes the best in its area.

The third policy investigated, called "locally complete", is to transmit an hypothesis if there is no more local KS processing that can be performed on the hypothesis. This condition is recognized when the acoustic region of an hypothesis "almost" covers the node's acoustic area-of-interest. This policy implements a simplified version of the locally complete strategy presented in Sec. 4.4.1. This version is simplified since the impact of a locally complete hypothesis is never explicitly evaluated. Rather, the successful extension of a phrase hypothesis to the boundaries of the node's area-of-interest is taken as an implicit indication that the hypothesis is important and should be transmitted. Additionally, in order to minimize the number of hypotheses transmitted, none of the intermediate phrase hypotheses used in the construction of a locally complete hypothesis are transmitted.

Due to the static allocation of the areas-of-interest and the small number of nodes (a maximum of three), a fully connected communication configuration was chosen. Thus, we are not able to

test more complicated and selective communication strategies in which a limited subset of nodes receives each transmission. In this broadcast strategy, all nodes receive the message, the sender does not receive a positive acknowledgment that the message has been received correctly, and the receiver does not know the identity of the sender.

5.4. Knowledge-Source Changes

Another aspect of the decomposition needing clarification is the changes made to the knowledge source configuration of the centralized system. The major change was adding the communication KSs. Additionally, several changes were required in previously existing KSs to remove implicit assumptions (sometimes very subtle) about the completeness of the information available at the time they execute. For example, the PREDICT KS, which uses syntactic knowledge to predict the set of words that might precede or follow a phrase hypothesis, uses the following heuristic:

If the number of words predicted in one direction is much smaller than the number in the other direction, predict only in the direction of the smaller number.

This heuristic attempts to apply the greatest constraint as soon as possible -- the assumption is that when the extended phrase(s) is (are) in turn extended, the added constraint of the initial extension will reduce the number of words subsequently predicted on the larger side. Since the verification of predicted words by the VERIFY KS is expensive, reducing the number of predicted words is highly desirable.¹

In the network system, however, this heuristic causes a problem because the node cannot extend the phrase in the desired direction if the acoustic information (at the lower, segment level) is outside its area-of-interest. We chose to handle this problem by modifying the heuristic to select a direction that can be verified locally, even if it means selecting the direction with the larger number of words, on the assumption that the added local processing is still better than introducing additional communication and possible redundant processing in some other nodes.

Additionally, some KS processing had to be modified to remove implicit assumptions about the sequential nature of hypothesis generation. For example, consider a blackboard that contains the two phrases "WHAT HAS" and "SMITH PUBLISHED IN 1974". Suppose that as a result of processing by the PREDICT and VERIFY KSs, the first phrase can be extended to the right to include the word "SMITH". The CONCAT KS, which performs this extension by creating the enlarged phrase hypothesis "WHAT HAS SMITH", also checks whether "SMITH" is the first word of an already existing phrase hypothesis, and in this case finds "SMITH PUBLISHED IN 1974". When it detects such a situation, it merges these two phrases, grammar permitting, into an enlarged phrase. In this case, the result is the complete phrase "WHAT HAS SMITH PUBLISHED IN 1974". This merging action is potentially very useful because it often eliminates the redundant computation involved in incrementally creating the merged phrase from the smaller one -- in this example, this would

¹The VERIFY KS also had to be modified so that it does not reject a word if there is insufficient data in the node's acoustic area-of-interest to make a valid decision.

involve creating the following sequence of phrases:

WHAT HAS SMITH
 WHAT HAS SMITH PUBLISHED
 WHAT HAS SMITH PUBLISHED IN
 WHAT HAS SMITH PUBLISHED IN 1974

Suppose, however, that the blackboard contains the phrase "HAS SMITH PUBLISHED IN 1974" instead of "SMITH PUBLISHED IN 1974". Now the CONCAT KS is unable to merge together the phrases, since the simple heuristic used for detecting merging situations does not handle the situation in which the phrases to be merged overlap by more than one word. In the sequential version of the Hearsay-II speech understanding system this situation never occurs because each time a phrase is enlarged by a word, the CONCAT KS checks for the possibility of merger (e.g., when either "HAS" is added to the end of "WHAT" or to the beginning of "SMITH PUBLISHED IN 1974") thus always detecting the possibility of merging. However, in a network system with CONCAT KSs operating asynchronously in parallel in separate nodes with incomplete and overlapping local blackboards, such situations often occur.

The issues posed by the CONCAT KS can be generalized as the following problem: how to avoid redundant computation caused by the generation (or reception) of information that overlaps or is subsumed by existing information. This problem does exist in sequential problem-solving systems but often can be minimized by employing simple heuristics with a global view of the problem-solving database -- the centralized version of the CONCAT KS does this. In distributed systems with incomplete and asynchronous processing, significant modification to local KSs may be required to handle this problem. These changes may also require communication of more detailed characteristics of the abstract information.¹

This instance of the problem was solved by modifying the RECEIVE KS. In brief, the following heuristic is used for merging externally received information on the blackboard:

The RECEIVE KS determines if that part of the received information totally contained in the node's area-of-interest already exists in whole or in part in the node. If so, this information is removed from the received hypothesis as long as the remaining part of the received hypothesis provides sufficient context for local KS processing to reconstruct the received hypothesis (by the incremental extension of the truncated version).

This heuristic attempts to decompose the received information so that the existing, centralized checks for redundant computation can be exploited.

For example, consider a two-node distributed Hearsay-II system attempting to recognize "WHAT HAS SMITH PUBLISHED IN 1974" (see Fig. 9). Consider the situation in which node 1 generates the one-word phrase "SMITH" and transmits it to node 2. Node 2 extends this phrase into "SMITH PUBLISHED", "SMITH PUBLISHED IN", and, finally, "SMITH PUBLISHED IN 1974". While node 2 is

¹In this experiment, more detailed characteristics of the underlying word support of a phrase hypothesis is transmitted with it in order to recognize whether the hypothesis is either redundant or subsumed under existing information in the local blackboard

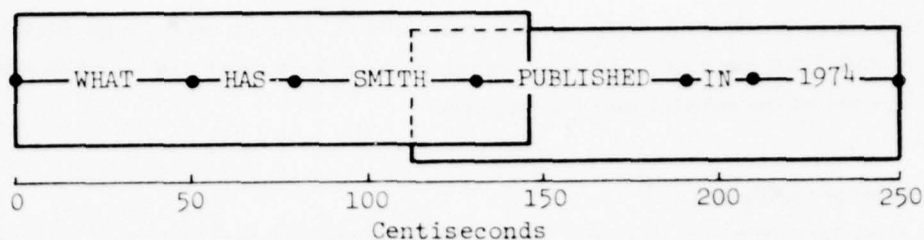


Figure 9: Location of words and areas-of-interest in two-node example.

performing these extensions, node 1 extends "SMITH" in the other direction into the phrases "HAS SMITH" and "WHAT HAS SMITH". Without the receiving heuristic, when either node received the extended phrase hypothesis from the other it would have to repeat its local processing to construct the complete sentence.

For example, when node 1 receives "SMITH PUBLISHED IN 1974", it would successively generate "HAS SMITH PUBLISHED IN 1974" and "WHAT HAS SMITH PUBLISHED IN 1974". However, with the receiving heuristic, node 1 would instead place on its blackboard the truncated phrase "PUBLISHED IN 1974"; this can be merged directly with "WHAT HAS SMITH" into the complete and correct phrase hypothesis.

While this receiving heuristic does not eliminate all redundant computation, combined with the locally complete transmission strategy it does avoid a significant amount of redundant computation in our test examples. The cost of this heuristic is the added computation to accomplish the incremental reconstruction and the delay in the effective use of information caused by the time required for the reconstruction.

These KS changes are not in themselves important. We describe them to give the reader a feeling for issues in organizing knowledge (i.e., algorithms and heuristics) for a distributed environment. These particular changes focus on the problems of processing with incomplete information and merging overlapping information. Both problems are caused by the asynchronous operation of the system. These same problems do occur in the centralized Hearsay-II system, because of the asynchronous interactions of KSs. However, in a centralized environment it is easier to build partial sequentializations which reduce these problems. And in some cases we did not even realize that we were building in such assumptions.

6. RESULTS

There are two main purposes for gathering experimental data on the performance of a network of Hearsay-II systems. The first is to provide empirical evidence for the assertion that the additional uncertainty introduced by distribution can be handled within the basic, uncertainty-resolving mechanisms of the Hearsay-II architecture. The second is to see if there are dynamic interaction phenomena among the nodes that we had not anticipated from our static

analysis, particularly phenomena dealing with communication bandwidth and overall performance. (It should be stated that some of KS changes discussed above were suggested by experimentation with early versions of the network Hearsay-II system.)

6.1. Network versus Centralized

The most important experimental results come from comparing the performance of a three-node Hearsay-II system with that of the centralized version. Given the requirements described in Sec. 5.2 and the lengths of the utterances in the test data, three nodes is about the maximum that can be used. Both systems were configured with the same task language (called "S5") which has a 250-word vocabulary and a very simple grammar.¹ We chose for test data a set of ten utterances that had been understood correctly by the centralized system.

The nodes in the network were configured with extensive overlap between their areas-of-interest (see Sec. 5.2). Figure 10 shows the ten sentences and the areas-of-interest for each of them. The locally complete strategy (see Sec. 5.3) was used for internode communication.

The network system correctly understood all ten of the utterances. Thus, the uncertainty introduced by this distribution of the problem-solving was handled by the basic Hearsay-II architecture without the need for additional mechanisms. This basic result has been substantiated by consistently-correct interpretations in several additional experiments with, in turn, 1) decreased area-of-interest overlaps, 2) less-constraining grammar, 3) alternative communication policies (Sec. 6.2), and 4) two-node configuration.

Figure 11 is a summary of the execution costs for running these ten utterances on the network system relative to the costs on the centralized system. The summary is along two dimensions: the processing time and the number of phrase hypotheses generated and transmitted. As described in Sec. 5.2, the selection of areas-of-interest for these experiments has led to a configuration in which all bottom-up processing through the word level can be accomplished with no internode communication. Since the purpose of these experiments is to investigate internode cooperation, as opposed to task-specific parallelism, the times reported are of the processing after that bottom-up phase has completed. Note that the results of the bottom-up phase are used throughout the subsequent processing -- in particular, the segment and word hypotheses within a node are constantly used by the node while investigating the extension of phrase hypotheses. The rationale of the distributed design is to avoid the transmission of the word and segment hypotheses to a central site. When reporting processing time in the network case, the time given is the maximum time over the three nodes, which is an estimate of the clock time of the simulated network.

¹The Hearsay-II Speech Understanding System is configurable with a varying range of task languages. The use of a simple language reduced the amount of computing resources required for the experimental runs.

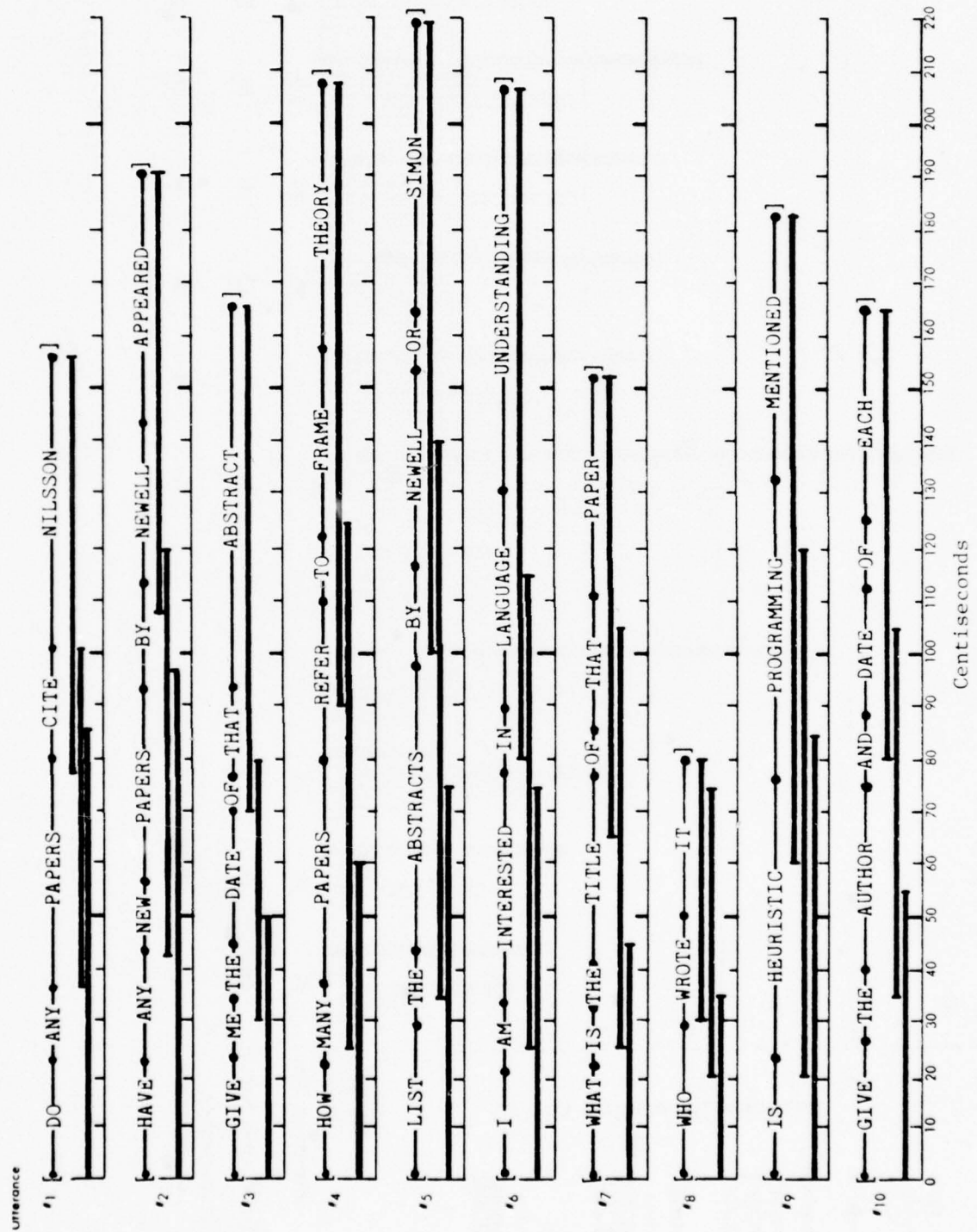


Figure 10: The test utterances and areas-of-interest.

(with "locally-complete" strategy)

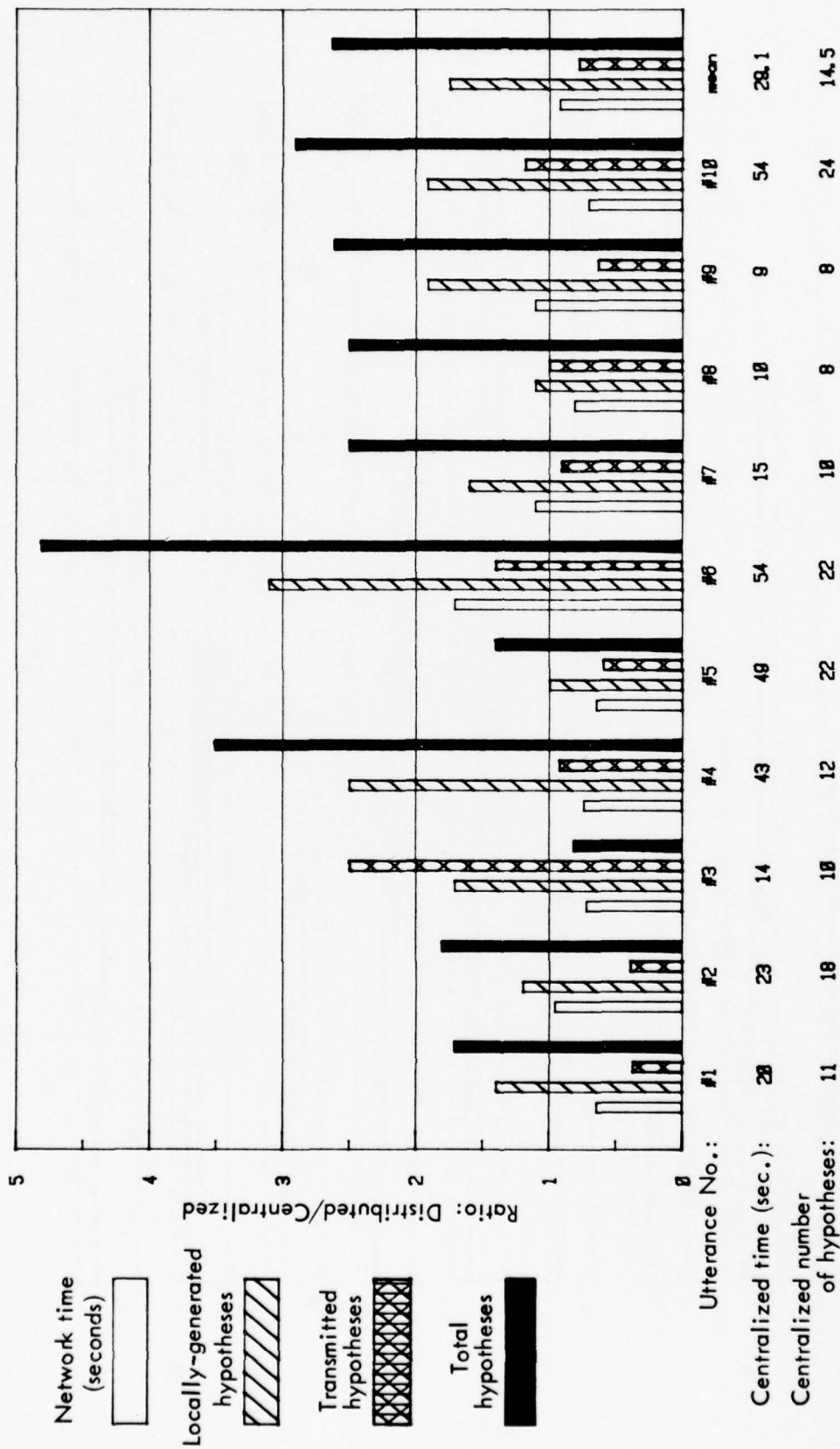


Figure 11: Performance of centralized vs. network systems.

For the network system, three counts of phrase hypotheses are used. First is the number of phrase hypotheses **generated locally** by each node, summed over the three nodes. This measures the amount of search more directly than does processing time. Next is the number of these hypotheses that were selected by the locally complete strategy for **transmission**. This is a measure of the channel costs for communication. Finally, there is the **total number** of phrase hypotheses that occurred; this is the sum over the three nodes of the number of hypotheses created locally by the node and the number of received hypotheses accepted by the node and placed on its blackboard.¹ For each of these three measures, Fig. 11 gives the ratio of that number to the number of hypotheses created in the centralized system.

The major conclusions that can be drawn from the summary statistics in Fig. 11 are:

- Effective cooperation was achieved among the nodes even though only 44% of the locally generated hypotheses were transmitted. This represents 77% of the number of hypotheses created in the centralized runs.
- There was a slight speedup of 10% in performing the interpretation above the word level with three nodes. Thus, the interpretation took 2.7 ($= 3 \times .9$) times as much processing as compared to the centralized version.

Recall that the times reported are of the high-level, highly cooperative processing only. If the bottom-up processing is included, which accounts for about half the time in the centralized system, there is an overall speedup of about 60% for the three-node configuration over the centralized version.

We classify the increase in the total amount of high-level processing into three areas: communication, incomplete information for knowledge application, and incomplete meta-information for focusing.

Communication costs include deciding which hypotheses to transmit and accept as well as the physical act of message passing. Also, the receiving node must merge accepted hypotheses into its blackboard structure. These sending and receiving functions account for about 6% of the processing time. To reduce the size of each message, the grammatical structure of the phrase hypothesis is not transmitted; rather, the receiving node recomputes that structure when needed, thus trading off additional processing for reduced communication bandwidth. None of these processing costs occur in the centralized system.

Incomplete information makes it more costly to process hypotheses. As discussed in Sec. 5.4, the heuristic in the PREDICT KS for selecting the direction of prediction was modified to be sensitive to the node's *area-of-interest*. The inability to predict in the direction of greater constraint leads to more word verification processing. A more subtle effect of a node's limited *area-of-interest* is a shift in the distribution of the length of phrase hypotheses towards hypotheses having fewer words. In general, shorter phrase hypotheses have less grammatical constraint on the number of

¹This third number may be more or less than the sum of the other two because a transmitted hypothesis is accepted by a receiving node only if it overlaps the node's *area-of-interest*. Thus, an hypothesis transmitted in a three-node network might be accepted by zero, one, or two nodes.

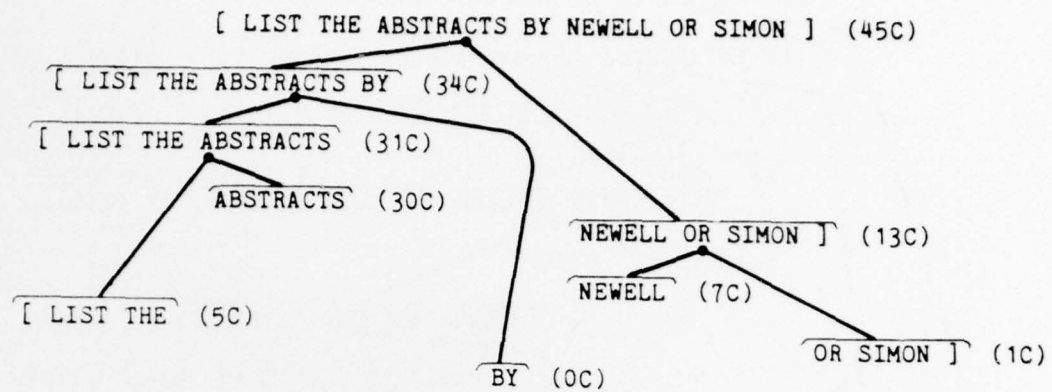
words they predict, which also leads to additional word verification. These effects showed up as a doubling of the number of words predicted per phrase hypothesis.

Incomplete meta-information can lead to redundant search and unnecessary search (i.e., with a low likelihood of a correct solution), which reduce the potential speed-up benefits of a parallel search. *Redundant search* occurs because there is no centralized scheduler to coordinate the search of nodes with overlapping areas-of-interest. As discussed in Sec. 5.4, we have added a mechanism to eliminate some of the redundant search caused by externally received hypotheses, but this mechanism itself has as its attendant cost the additional computation to reconstruct received information. *Unnecessary search* occurs because the search paradigm is opportunistic across the length of the utterance, i.e., working out from a few islands of reliability discovered in the data. These islands are not, in general, distributed uniformly among the nodes in the network. This leads to cases in which a particular node can do little effective processing until it receives constraining information, i.e., a reliable island, from another node. Likewise, after a node has fully explored all of its reliable islands, it may also have little effective processing to do. The processing that occurs before the node receives a reliable island and the processing after it has fully exploited all of its reliable islands is, from a global view, unnecessary. Thus, the opportunistic scheduling partially sequentializes the search. The effect this has on the parallel speed-up in a network system depends on the distribution of islands across the nodes -- the more uniform the distribution, the greater the speed-up. Figure 12 illustrates this by showing how one of the test utterances was recognized in the centralized and distributed systems.

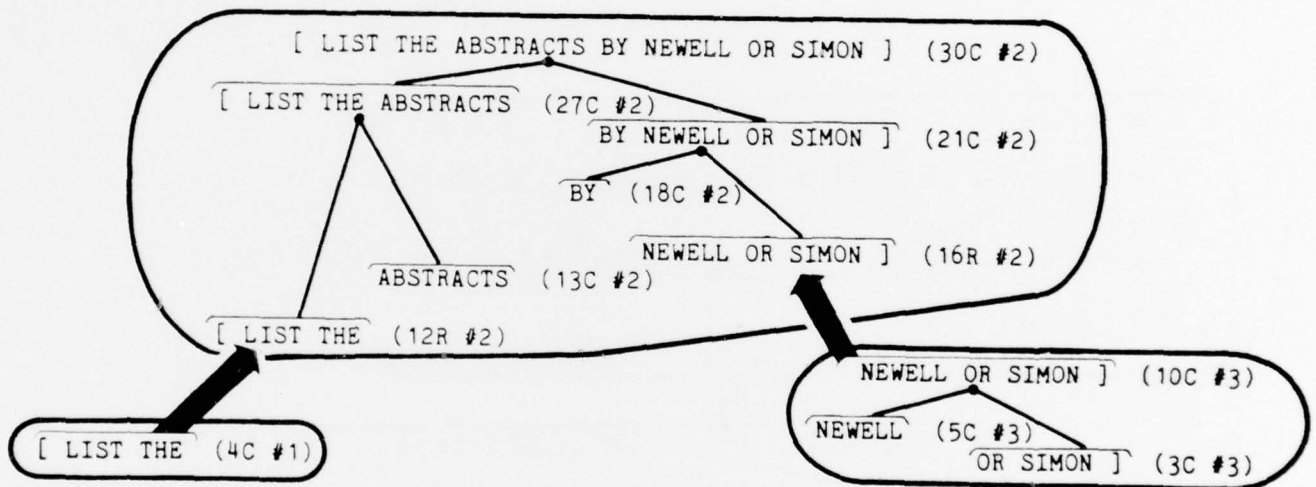
Because of the uncertainty in knowledge and data in speech understanding, such unnecessary search may produce hypotheses with sufficient credibility and scope to be transmitted. This internode communication is itself unnecessary and may distract nodes doing productive work, thus causing even more unnecessary search. This distraction occurs because the estimate of impact of an hypothesis is based in part on its scope (length). Thus, a long, moderately rated hypothesis may be considered to have more impact than a short, highly rated one. If a node lacking a reliable island does not soon receive constraining hypotheses, it is often able to develop hypotheses of moderate credibility and large scope which it then transmits. If such an hypothesis is received by a node with a highly reliable island before it has been able to develop that island fully, the node may switch its attention to the longer, received hypothesis, thus delaying, perhaps indefinitely, the useful processing of the shorter, highly credible island. The recognition trace of the utterance shown in Fig. 13 shows the results of such distraction.

This method of estimating impact for focusing decisions is reasonable in a centralized system in which all the input data are received together. In such a system, the development of hypotheses is implicitly more synchronized -- the highly rated island would have been extended before the lower rated hypothesis would have been developed. A possible solution to this problem in the network system is to normalize the estimate of impact of received hypotheses according to the scope of the largest locally generated ones.¹

¹It might be desirable to expand such differential treatment of received hypotheses, e.g., to use meta-information about the transmitting node for evaluating the received hypothesis.

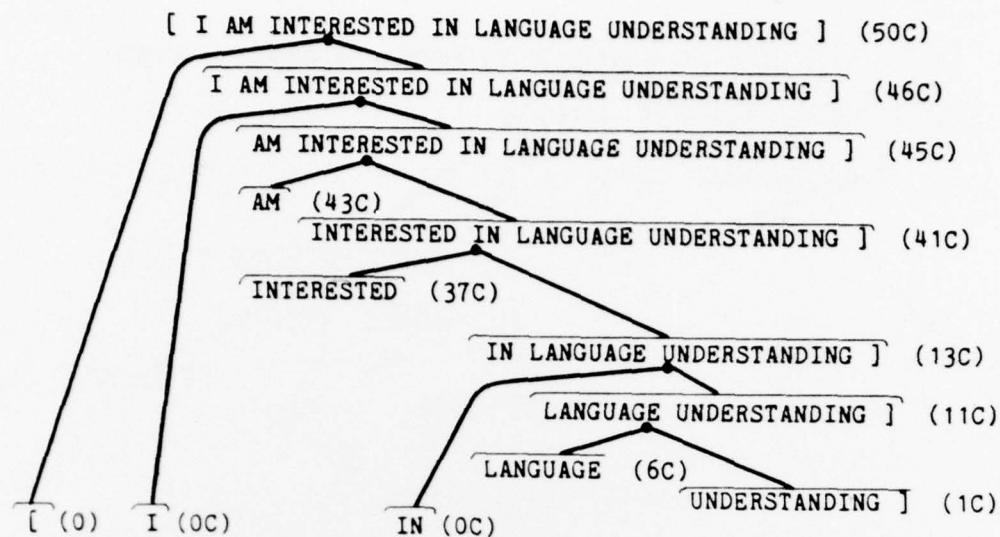


a: In the centralized system.

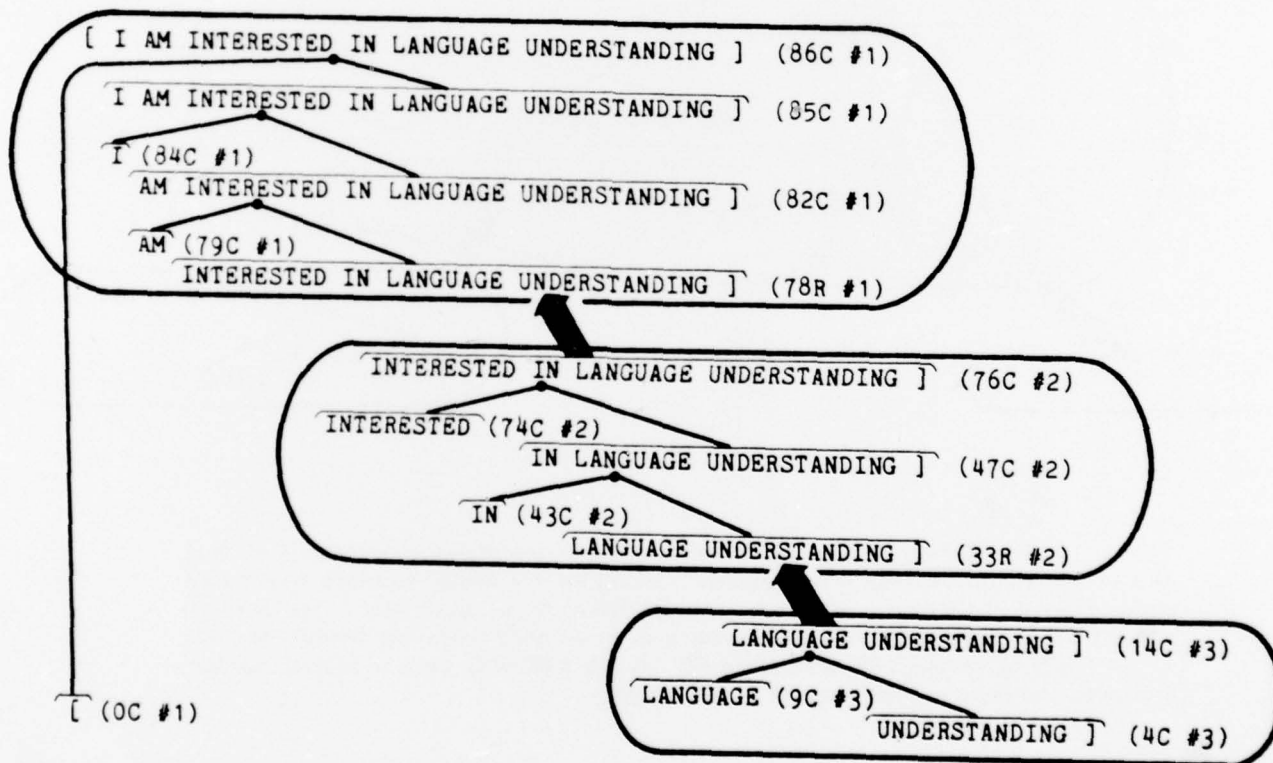


b: In the 3-node configuration.

Figure 12: Recognition process for those partial interpretations of utterance #5 that led to the correct overall interpretation. Joined lines indicate intranode hypothesis creation. Arrows show internode communication of a hypothesis. Numbers in parentheses indicate network processing time in seconds when the hypothesis was created (C) or received as a message (R). In the multinode case, a second number indicates the node number (e.g., #2 for node two).



a: In the centralized system.



b: In the 3-node configuration.

Figure 13: Recognition process for those partial interpretations of utterance #6 that led to the correct overall interpretation.

Five utterances were also run using a more complex (i.e., less constraining) grammar, called "S15". Again, all five were recognized by both the centralized and three-node configurations, adding credence to our hypothesis that the accuracy of the problem-solving can be maintained within the distributed configuration. In these runs, the overall speedup increased to 30% from the 10% of the simpler grammar, indicating more parallelism in the larger search space. The fraction of hypotheses transmitted remained similar to the fraction in the simpler grammar runs.

6.2. Transmission Policies

The network data in the previous section were generated using the locally complete transmission policy. Figure 14 presents experimental comparisons of that policy with those of dynamic thresholding and full transmission. (See Sec. 5.3 for descriptions of these policies.) The utterances used were the first five of the ten used in the previous section; the same areas-of-interest were used. All five utterances were correctly understood under all three transmission policies.

On the basis of both processing time and number of hypotheses transmitted, locally complete is more efficient than the dynamic thresholding, which in turn is better than the full transmission. It thus appears that the timeliness advantage of the dynamic thresholding policy is dominated by the reductions in redundant processing and distracting communication of the locally complete. In some experiments with a more complex grammar, the differential between the two selective policies was reduced -- our conjecture is that the extra timeliness of the dynamic thresholding policy becomes more important as the complexity of the search increases.

6.3. Communication with Errors

In order to assess the robustness of the network system with respect to communication errors, experiments were run in which messages received by a node are randomly discarded with a specified probability. This models errors in communication systems that have good error detection but poor correction capabilities, e.g., packet radio. Selection at the receiving end allows for cases in which a broadcast message is received successfully by some nodes but not others.

Two characteristics of the network system should make it robust in the face of communication errors. First, there are redundancies that can recreate the information in lost messages; second, the system can exploit the recreated information even though it arrives later than would have the original, lost communication. There are several ways of recreating the lost information:

- The overlapping of areas-of-interest leads to the possibility of creating redundant information directly.

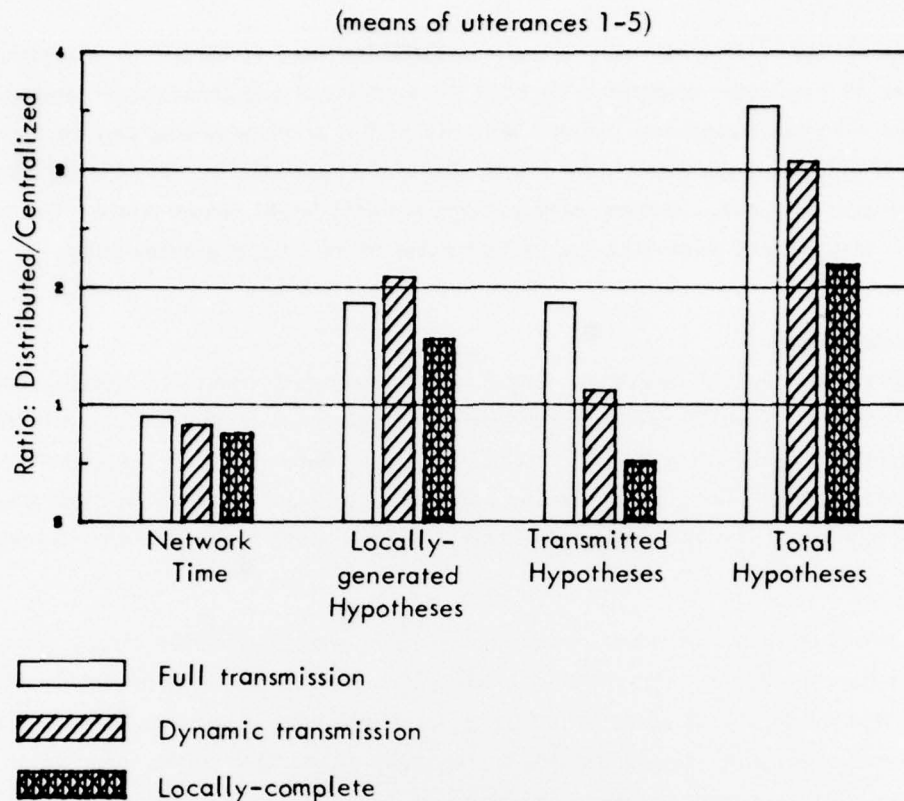


Figure 14: Performance comparisons of the three transmission policies.

- The transmission policy can introduce redundant communications. For example, the dynamic threshold policy (and the full transmission policy) can produce a sequence of messages representing the stages of development of a partial solution. Each message in the sequence subsumes the information in the previous messages. This redundancy does not exist in the locally complete policy, which transmits only the final message in that sequence; it is for this reason that dynamic threshold was used for this experiment. Other mechanisms, such as murmuring (Sec. 4.4), can be used for additional explicit redundant communication -- we have not explored them in these experiments.
- The broadcasting of messages makes it possible that messages might be lost to one node but received by another. The node that correctly receives the information might operate on it and subsequently broadcast a message based on information in the original message. The rebroadcast may be received by the node that lost the original version. This propagation of information among the nodes thus implicitly creates redundant communication paths.
- The method of building an interpretation by incremental aggregation of partial

interpretations makes it possible to derive a correct interpretation in multiple ways. This kind of behavior has been observed in the centralized version of the Hearsay-II speech understanding -- for example, cases have occurred in which a complete interpretation could not be constructed from one correct island of reliability because of KS errors but could be derived from another. Because a particular message may not be crucial for all ways of deriving a correct interpretation, its loss does not preclude a correct interpretation.

These experiments used the same data as those in Sec. 6.2. The dynamic threshold transmission policy was used, to provide more redundancy in communication than the locally complete. Figure 15 shows the performance with 0%, 25%, 35%, and 50% of the messages discarded. One utterance of the five was not correctly recognized (i.e., no complete interpretation was constructed in the maximum allocated processing time) in the 25% and 35% cases, and three were missed in the 50% case. There are several interesting points about the statistics. For example, the execution times for a number of runs was decreased *because of* the errorful communication channel. This occurred when messages discarded due to the simulated communication failures happened to be either *incorrect or redundant*. Other runs, as expected, required additional processing time and communication to recreate the nonredundant information lost due to communication failure.

Several runs were not correctly recognized because a message was lost which contained the first or last word in the utterance.¹ Information about these extreme areas is contained in only a single node and is thus especially difficult to recreate in another node. The loss of this information is not always fatal. Figure 16b shows an example where first-word information was lost on two separate transmissions ([+HAVE+ANY from node 1 to 2, and [+HAVE+ANY+NEW+PAPERS+BY from node 1 to 3). The system, however, was resilient enough to recreate the information through a round-about path. Figure 16a is a trace of the system recognizing the utterance when this information was not lost.

In summary, the system's performance with a faulty communication channel lends credence to our belief that the architecture is resilient and permits a tradeoff between the amount of processing and reliability of communication. We further believe that the introduction of a knowledge-based murmuring scheme would correct most of the incorrect runs without increasing communication costs significantly.

7. CONCLUSIONS

Let us review our model for distributed interpretation systems:

There is a network of systems (nodes), each of which is able to perform significant local processing in a self-directed way. For example, if a node does not receive a particular piece of information in a given amount of time, it is able to continue processing, using whatever information is currently available to it.

¹Because of the randomness of message lossage, this happened to occur in utterance #2 in the 25% error case but not in the 35% case.

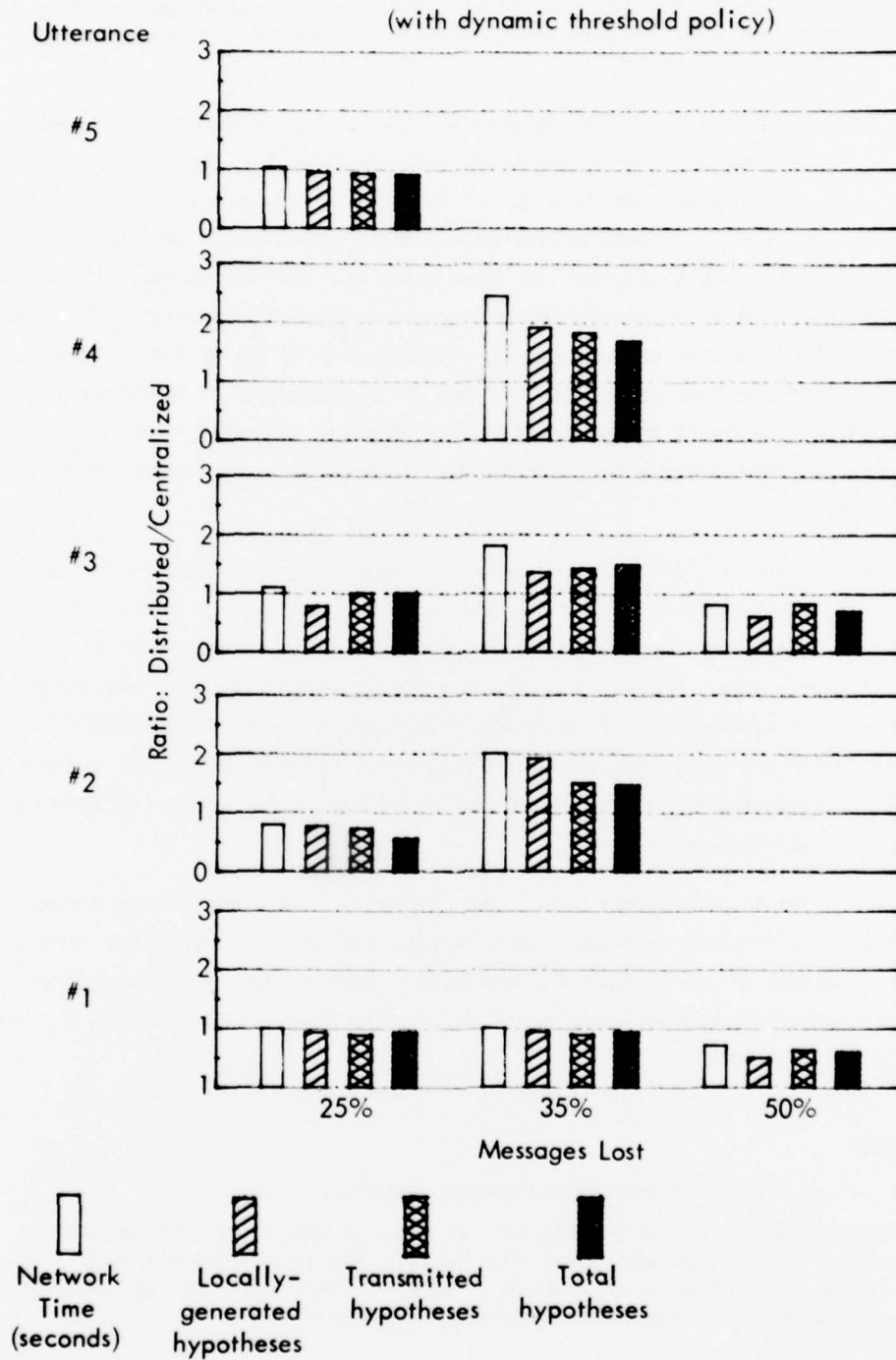
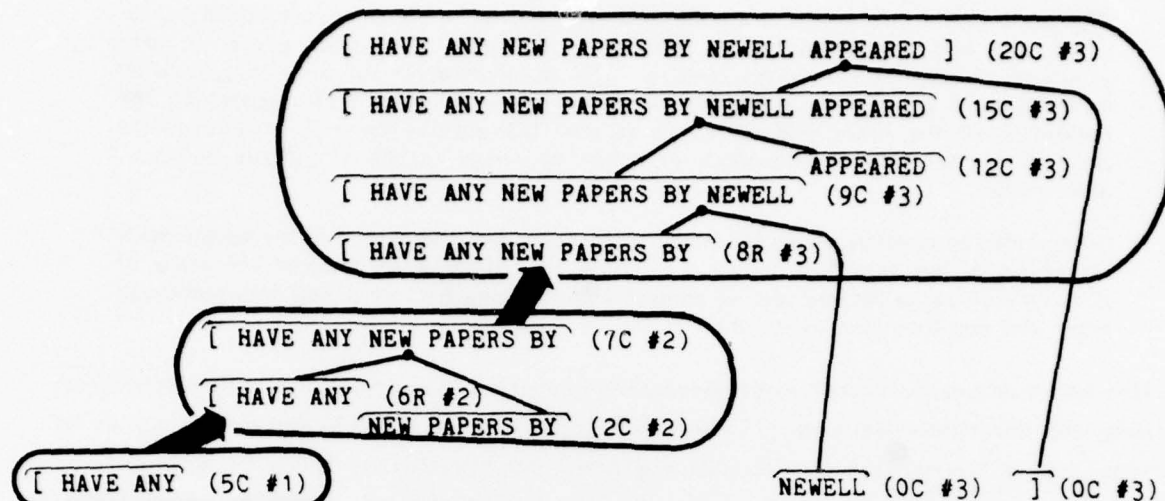
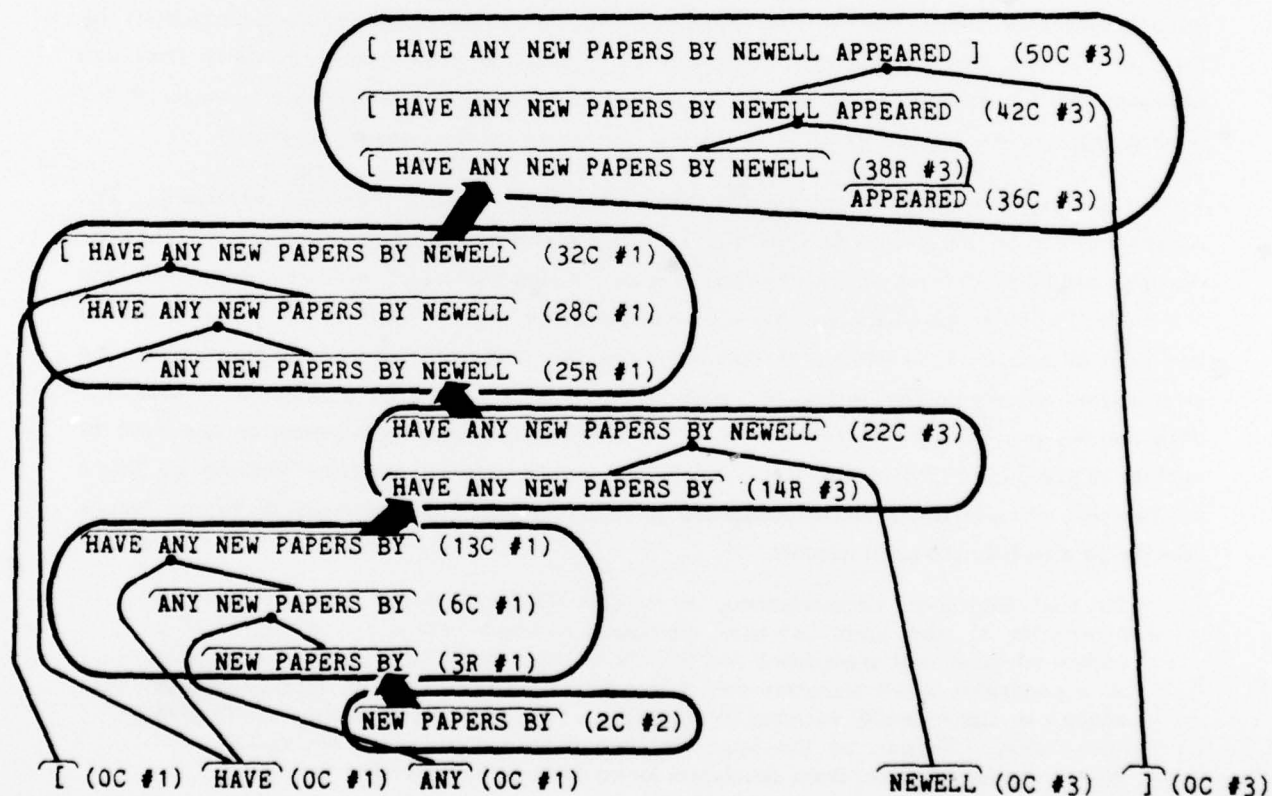


Figure 15: Performance with communication losses.



a: With no messages discarded.



b: With 35% of the messages discarded.

Figure 16: Trace of utterance #2 processing with and without messages discarded, showing those partial interpretations that led to the correct overall interpretation.

The parts of the problem that a node is responsible for working on is called its *area-of-interest* and is defined by the information it needs and produces. In general, areas-of-interest of the nodes overlap. The local database of a node (i.e., what information it actually has) may be incomplete or inconsistent with respect to the databases of the other nodes. Nodes resolve the uncertainty in their information through an iterative, asynchronous exchange of partial results at various levels of abstraction.

Control of cooperation among the nodes is decentralized and implicit in the autonomous behaviors of the individual nodes. Each node uses its local estimate of the state of problem-solving in the network to control its processing (i.e., what new information to generate) and transmissions to other nodes.

This model differs from conventional approaches to distributed system design in its emphasis on dealing with uncertainty and error in control, data, and algorithms caused by the distribution as an integral part of the network problem-solving process. An attractive structure for accomplishing this is an opportunistic problem-solving structure and, in particular, one which has implicit (data-directed) information flow and control flow.

The conventional approach to the design of distributed systems is to overlay some basic, centralized problem-solving strategy with new mechanisms that handle the uncertainty and errors introduced by the distribution. It is our hypothesis that this conventional approach limits both the type of systems that can be distributed effectively and the environments in which they can operate. We feel the key to the design of distributed systems is to incorporate mechanisms for dealing with uncertainty and error as an integral part of the problem-solving approach.

The Hearsay-II architecture appears to be a good one for this integrated approach. The processing can be partitioned or replicated naturally among network nodes because it is already decomposed into independent, self-directed modules (i.e., the KSs) which interact anonymously and are limited in the scope of the data they need and produce. Issues involved in the distribution of the control and data structures of Hearsay-II can be dealt with effectively because of the mechanisms already in the system for resolving uncertainty caused by incomplete or incorrect data and KS processing. Figure 17 reviews these mechanisms and their impact on the ease of system distribution.¹ Within the basic distributed problem-solving structure defined by these mechanisms, several other mechanisms have been incorporated or proposed to handle issues specific to a distributed environment:

- To limit internode communication, an incremental transmission mechanism (with processing at each step) has been developed in which only a limited subset of a node's information is transmitted and to only a limited subset of nodes. A node acts as a generator which transmits only a few most credible pieces of information and which can subsequently respond to stagnation of progress by producing alternative information. As part of this approach, two policies ("dynamic thresholding" and "locally complete") have been developed for controlling the generator function.

¹Not all these mechanisms were exploited in the distributed Hearsay-II speech understanding system described in the previous section. In general, the possibility for exploiting a particular mechanism is dependent on the specifics of the problem-solving application being distributed.

Mechanism: Opportunistic nature of information gathering -- Problem-solving is viewed as an incremental, opportunistic, and asynchronous process in which decisions, if they look promising, can be made with incomplete information and later re-evaluated in the light of new information.

Mechanism: Use of abstract information -- Because the problem-solving database is structured as a loose hierarchy of increasingly more abstract problem representations, an abstract representation of one aspect of the solution can be used to constrain analysis of other aspects of the problem.

Mechanism: Incremental aggregation -- A solution is constructed through the incremental piecing together of mutually constraining and consistent information; incorrect partial solutions naturally die out as a result of this process.

Mechanism: Problem-solving as a search process -- Because of uncertainty in data and KS processing, many alternative partial solutions need to be examined in the process of constructing a complete and consistent solution; in this search process, the more uncertainty there exists, the larger the number of alternatives that, in general, have to be explored.

Mechanism: Functionally-accurate definition of solution -- Due to the opportunistic nature of processing and the existence of diverse and overlapping KSs, the correct solution may be derivable in different ways, i.e., using different ordering sequences for incrementally constructing the solution components or using different solution components. Because a solution is based on a set of mutually constraining pieces of information, it is also possible for a correct solution to incorporate information that is correct but not considered very likely, or to use incorrect information that is considered very likely.

Impact: Reduced need for synchronization -- Because of this style of problem-solving, a node does not have an a priori order for processing information and can exploit incomplete local information. Thus, the processing order within nodes and the transmission of information among nodes does not need to be synchronized.

Impact: Reduced internode communication bandwidth -- The ability to use abstract information permits nodes to cooperate by using messages with high information content; thus, the communication bandwidth needed for effective cooperation is reduced.

Impact: Automatic error detection -- This method of problem-solving allows a distributed system to detect and reduce the impact of incorrect decisions caused by incomplete and inconsistent local databases and communication losses.

Impact: Internode parallelism -- The requirement that many alternative partial solutions need to be examined generates the possibility that this search can be carried out in parallel by different nodes. The asynchronous nature of information gathering introduces the possibility for additional parallelism, since different aspects of the problem and different information levels can be worked on independently. Further, the introduction of additional uncertainty through incomplete and inconsistent local databases can be traded off against more search -- to the degree that this extra search can be done in parallel and does not itself generate proportionally more internode communication, internode bandwidth can be lowered without significant degradation in system response time.

Impact: Self-correcting -- Because there are multiple paths from which a solution can be derived, it is possible to correct for what would be considered fatal errors in a conventional distributed problem-solving system. Additionally, system reliability can be varied without modifying the basic problem-solving structure, through the appropriate selection and focusing of local node processing. For example, it is possible to improve reliability by enlarging the overlap among nodes' areas-of-interest, thus increasing the likelihood of generating redundant information. This increases the number of alternative ways that a solution can be derived.

Figure 17: Hearsay-II mechanisms and their impacts on distributed systems.

- To increase network reliability, a knowledge-based mechanism called "murmuring" has been proposed. Here, a node retransmits high-impact information if during a specified time interval it neither receives nor generates higher-impact information. Murmuring can be used to correct for lost communications due to intermittent channel or node failures and to bring new or moving nodes up-to-date.
- To guarantee the appropriate communication connectivity among nodes, a decentralized mechanism for constructing a communication network has been developed. Using this mechanism, which relies on descriptions of the I/O characteristics of each node, nodes act as store-and-forward message processors to provide needed connectivity. A similar mechanism can be used for dynamic allocation of processing tasks among nodes.
- To provide more sensitive implicit internode control while still retaining decentralization, each node may transmit explicitly its local control information ("meta-information"). Nodes can thus determine more directly the state of processing in other nodes.

The experiments described here explore these mechanisms in only a limited way. A number of issues need to be resolved in order to gain an understanding of the more general applicability of this approach:

Distributed Focus of Control

- How to coordinate in a decentralized and implicit way the activity of nodes that have overlapping (i.e., redundant) information, so as to control redundant computation.
- How to decide locally that a node is performing unnecessary computation and how to select the aspect of the problem on which it should instead focus its attention. This is the problem of dynamic allocation of information and processing capabilities of the network.¹

Self-correcting Computational Structure

- What and how much uncertainty (errors) can be handled using these types of computational structures, and what is the cost in processing and communication to resolve the various types of errors?

Task Characteristics and the Selection of an Appropriate Network Configuration

- What characteristics of a task can be used to select a network configuration appropriate for it? When can implicit control and information flow structures be used? Similarly, when should flat, hierarchical, or matrix configurations, or mixtures of them, be used? Candidate characteristics include the patterns of KS interaction, the type, spatial distribution, and degree of uncertainty in information, interdependencies of partial interpretations, size of the search space, desired reliability, accuracy, responsiveness, and throughput, and available computing resources.

The Hearsay-II speech understanding system, with only minor changes, performs well as a cooperating network even though each node has a limited view of the input data. In the

¹This issue is related to the classical allocation problem in networks: How to decide if the cost of accessing a distant database is too expensive and whether, instead, the processing should be moved closer to the data or the data moved closer to the processor.

experiment with errorful communication, system performance degrades gracefully with as much as 50% of the messages lost; this experiment also indicates that the system can often compensate automatically for the lost messages by performing additional computation. These results support our general model of distributed systems design. They also indicate that the Hearsay-II architecture is a good one to use as a basis for this approach.

ACKNOWLEDGMENTS

We wish to thank James Adams for his help with the programming of the multinode simulation. Also, helpful comments on various drafts of this report were generously given by Jeff Barnett, Daniel Corkill, Randy Davis, Allen Hanson, Rick Hayes-Roth, Jasmina Pavlin, Daniel Schwabe, and Yechiam Yemini.

REFERENCES

- [Barrow 76] Barrow, H. G. and J. M. Tennenbaum, *MSYS: A System for Reasoning About Scenes*, SRI International, AI Center, Report 121, April 1976.
- [Baudet 76] Baudet, G. M., *Asynchronous Iterative Methods for Multiprocessors*, Carnegie-Mellon University, Computer Science Department, November 1976.
- [Drazovich 78] Drazovich, R. J., and S. Brooks, "Surveillance Integration Automation Project (SIAP)," *Distributed Sensor Nets Workshop*, Carnegie-Mellon University, December 1978, pp. 119-121.
- [Engelmore 77] Engelmore, R. S., and H. P. Nii, *A Knowledge-based System for the Interpretation of Protein X-ray Crystallographic Data*, Stanford University, Computer Science Department, Stan-CS-77-589, 1977.
- [Erman 75] Erman, L. D., and V. R. Lesser, "A Multi-level Organization for Problem Solving Using Many Diverse Cooperating Sources of Knowledge," *Proc. 4th International Joint Conference on Artificial Intelligence*, Tbilisi, USSR, 1975, pp. 483-490.
- [Erman 79] Erman, L. D., and V. R. Lesser, "The Hearsay-II System: A Tutorial," W. A. Lea (ed.), *Trends in Speech Recognition*, Prentice-Hall, Englewood Cliffs, NJ, 1979, Chap. 16.
- [Fennell 77] Fennell, R. D., and V. R. Lesser, "Parallelism in AI Problem-Solving: A Case Study of Hearsay-II," *IEEE Trans. on Computers* C-26 (February 1977), 98-111.
- [Galbraith 73] Galbraith, J., *Designing Complex Organizations*, Addison-Wesley, New York, 1973.
- [Hanson 78] Hanson, A. R., and E. M. Riseman, "VISIONS: A Computer System for Interpreting Scenes," A. Hanson and E. Riseman (ed.), *Computer Vision Systems*, Academic Press, New York, 1978, pp. 303-333.
- [Hayes-Roth 77a] Hayes-Roth, F. and V. R. Lesser, "Focus of Attention in the Hearsay-II System," *Proc. 5th International Joint Conference on Artificial Intelligence*, Cambridge, Mass., 1977, pp. 27-35.
- [Hayes-Roth 77b] Hayes-Roth, F., "The Role of Partial and Best Matches in Knowledge Systems," D. A. Waterman and F. Hayes-Roth (ed.), *Pattern-Directed Inference Systems*, Academic Press, New York, 1977.

- [Lesser 75] Lesser, V. R., R. D. Fennell, L. D. Erman, and D. R. Reddy, "Organization of the Hearsay-II Speech Understanding System," *IEEE Trans. on Acoustics, Speech, and Signal Processing* 23 (1975), 11-23.
- [Lesser 77] Lesser, V. R., and L. D. Erman, "A Retrospective View of the Hearsay-II Architecture," *Proc. 5th International Joint Conference on Artificial Intelligence*, Cambridge, Mass., 1977, pp. 790-800.
- [Lesser 78] Lesser, V. R., and D. D. Corkill, *Cooperative Distributed Problem Solving: A New Approach for Structuring Distributed Systems*, University of Massachusetts, Department of Computer and Information Sciences, Report COINS 78-7, May 1978.
- [Lesser 79] Lesser, V. R., J. Pavlin, and S. Reed, *Analysis of Models for Distributed Interpretation Systems*, University of Massachusetts, Department of Computer and Information Sciences, 1979. (to appear)
- [Lowerre 79] Lowerre, B. T. and R. Reddy, "The HARPY Speech Understanding System," W. A. Lea (ed.), *Trends in Speech Recognition*, Prentice-Hall, Englewood Cliffs, NJ, 1979, Chap. 15.
- [Nii 78] Nii, H. P. and E. A. Feigenbaum, "Rule-based Understanding of Signals," D. A. Waterman and F. Hayes-Roth (ed.), *Pattern-Directed Inference Systems*, Academic Press, New York, 1978.
- [Rosenfeld 76] Rosenfeld, A., R. A. Hummel, and S. W. Zucker, "Scene Labeling by Relaxation Operators," *IEEE Trans. on Systems, Man and Cybernetics* SMC-6 (1976), .
- [Simon 62] Simon, H. A., "The Architecture of Complexity," *Proc. Amer. Philosophical Soc.* 106 (1962), 467-482.
- [Smith 78] Smith, R. G., and R. Davis, *Distributed Problem Solving: The Contract Net Approach*, Stanford University, Computer Science Department, HPP-78-7, Stan-CS-78-667, September 1978.